

Технически университет – Варна

Факултет: ФИТА

Катедра: Софтуерни и интернет технологии (СИТ)

Специалност: СИТ

Протокол №1

Тема на проекта:

Влогове, вариант 4

Изготвил:

Име: Теодор Николаев Николов

Факултетен номер: 20621504

20621504 ТЕОДОР НИКОЛАЕВ НИКОЛОВ

Влогове

Да се напише компютърна програма, реализираща информационна система, която поддържа финансови влогове. Програмата съхранява и обработва данни за вложителите (Номер на сметка, Име, списък с влогове, откриване на сметка) в банка. Максималният брой вложители е 50. Всеки вложител може да има от 1 до 3 влога (валута, стойност) с различни валути.

Базова задача – Сложност ниска

- A. Меню за избор на функциите в програмата. (7 седмица)
- B. Добавяне на вложители (7-8 седмица)
 - a. Добавяне на нов вложител с един влог. Избира се валутата на влога (BGN (лев), USD (долар), EURO (евро));
 - b. Добавяне на списък от вложители. Въвежда се цяло число *n* и след него *n* на брой вложители с един влог. Избира се валутата на влога (BGN (лев), USD (долар), EURO (евро)).
- C. Извеждане на всички вложители на екрана (8-9 седмица)
 - a. Извеждане на вложители с влог BGN
 - b. Извеждане на вложители с максимален влог в EURO

Допълнение Първо – Сложност средна (+ Базова задача)

- D. Добавяне на нов влог: (9-10 седмица)
 - a. Всеки вложител (по номер на възложител) с по-малко от 3 влога може да му се създаде нов влог за валута по която няма влог с минимална стойност 500;
 - b. Ако не съществува вложител с този номер да се изведе подходящо съобщение;
 - c. При опит за добавяне на влог на вложител с 3 влога се изписва подходящо съобщение;

Допълнение Второ – Сложност висока (+ Базова задача + Допълнение Първо)

- E. Изтегляне на влог (10-11 седмица)
 - a. Въвеждат се номер на вложител, избира се влога от който ще се тегли и се изтегля цялата или част от сумата по влога.
 - b. Въвежда се номер на вложител и се извличат всичките му влогове, а профила му се закрива;

Допълнение Трето – Сложност висока (+ Базова задача + Допълнение Първо + Допълнение Второ)

- F. Справки за влоговете с под меню (11-12 седмица)
 - a. Извеждане на вложителите в подреден ред по:
 - i. Датата на основаване на влоговете;
 - ii. Общата стойност на влоговете.
 - b. Извеждане на вложителя с най-голям влог от въведена валута;
 - c. Извеждане на вложителите, които имат само един влог.
- G. Данните в програмата да могат да се запазват във файл между две стартирания на програмата.

Допълнение Четвърто – (за допълнителни точки)

- H. Допълнителни условия:
 - a. За точка B при добавянето на един вложител да се даде възможност, да се избере дали да се въведе нов или да се прекъсне въвеждането.
 - b. За точка C да се реализира изписването по 5 вложителя на страница. Пример: Има въведени 30 вложители. Извеждат се 5 вложителя, избира се следваща страница, извеждат се следващите 5 вложителя и т.н.
 - c. За точка G да се записват данните в двоичен файл.

Анализ на решението

Функции

```
int menu();  
void add_depositor(person p[],int& i);  
void print_depositor(person p[],int i);  
void add_deposit(person p[],int i);  
void withdrawal_deposit(person p[],int& i);  
void deposits(person p[],int h);  
void references(person p[],int i);  
void sort_depositors(person p[],int i);  
void max_deposit(person p[],int i);  
void depositors_one(person p[],int i);  
void save(person p[],int i);  
void fileToProgram(person p[],int& i,int& FP);  
void file();  
void clear_file();
```

A. Меню за избор на функциите в програмата

Функция: int menu();

B. Добавяне на вложители

Функция: void add_depositor(person p[],int& i);

C. Извеждане на всички вложители на екрана

Функция: void print_depositor(person p[],int i);

D. Добавяне на нов влог

Функция: void add_deposit(person p[],int i);

E. Изтегляне на влог

Функция: void withdrawal_deposit(person p[],int& i);

Под функция: void deposits(person p[],int h);

F. Справки за влоговете с под меню

Функция: void references(person p[],int i);

Под функция: void sort_depositors(person p[],int i);

Под функция: void max_deposit(person p[],int i);

Под функция: void depositors_one(person p[],int i);

G. Данните в програмата да могат да се запазват във файл между две стартирания на програмата.

Функции: извършват различни действия с файла.

void save(person p[],int i); записване на информацията

void fileToProgram(person p[],int& i,int& FP); прехвърляне на информацията

void file(); показване на информацията от файла

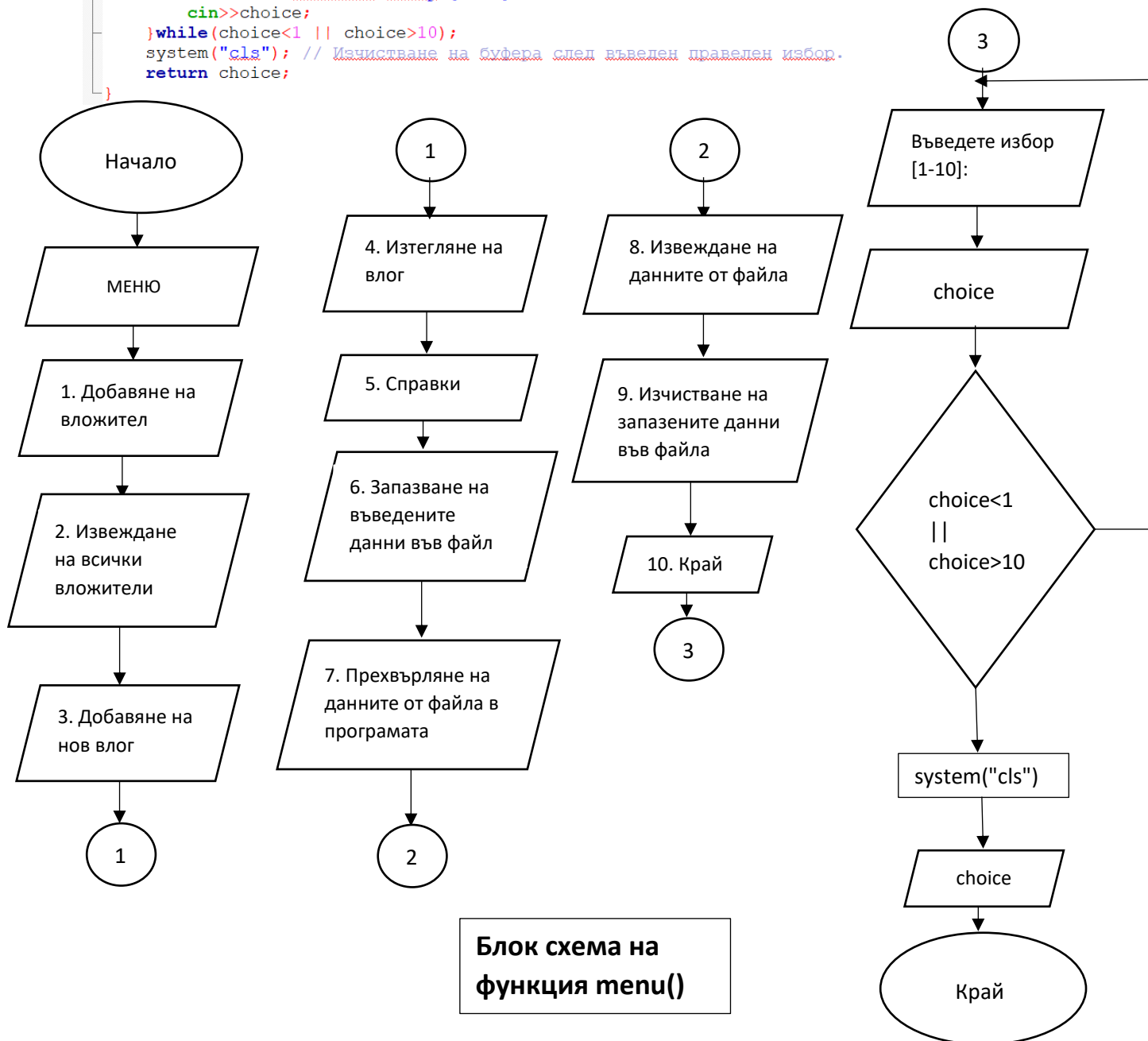
void clear_file(); изчистване на файла

1. Базова задача:

А) Меню за избор на функциите в програмата:

За изпълняването на тази част от задачата създавам функцията `menu()`, която е от тип `int`, тъй като връща стойност(число)/изборът, който сме направили от менюто. Функцията не използва входни данни. Функцията е съставена от `cout`-е, които представят менюто, и от `do-while` цикъл, който следи за правилно въведен избор от 1 до 10.

```
int menu()
{
    int choice;
    cout<<"\n\t МЕНЮ";
    cout<<"\n 1. Добавяне на вложител";
    cout<<"\n 2. Извеждане на всички вложители";
    cout<<"\n 3. Добавяне на нов влог";
    cout<<"\n 4. Изтегляне на влог";
    cout<<"\n 5. Справки";
    cout<<"\n 6. Запазване на въведените данни във файл";
    cout<<"\n 7. Прехвърляне на данните от файла в програмата";
    cout<<"\n 8. Извеждане на данните от файла";
    cout<<"\n 9. Изчистване на запазените данни във файла";
    cout<<"\n 10. Край"<<endl;
    do{ // С помощта на цикъл въвеждаме нашия избор, който задължително трябва да е от 1-10
        // ако не е в този диапазон трябва потребителя да въведе нова стойност.
        cout<<"\n Въведете избор [1-10]:";
        cin>>choice;
    }while(choice<1 || choice>10);
    system("cls"); // Изчистване на буфера след въведен правилен избор.
    return choice;
}
```



Упътване за употреба на подточка А

Както се вижда на конзолата, при въвеждане на число извън диапазона[1-10] се пита за ново въвеждане и така докато не се въведе число в този диапазон.

```
МЕНЮ
1. Добавяне на вложител
2. Извеждане на всички вложители
3. Добавяне на нов влог
4. Изтегляне на влог
5. Справки
6. Запазване на въведените данни във файл
7. Прехвърляне на данните от файла в програмата
8. Извеждане на данните от файла
9. Изчистване на запазените данни във файла
10. Край

Въведете избор [1-10]:12

Въведете избор [1-10]:1_
```

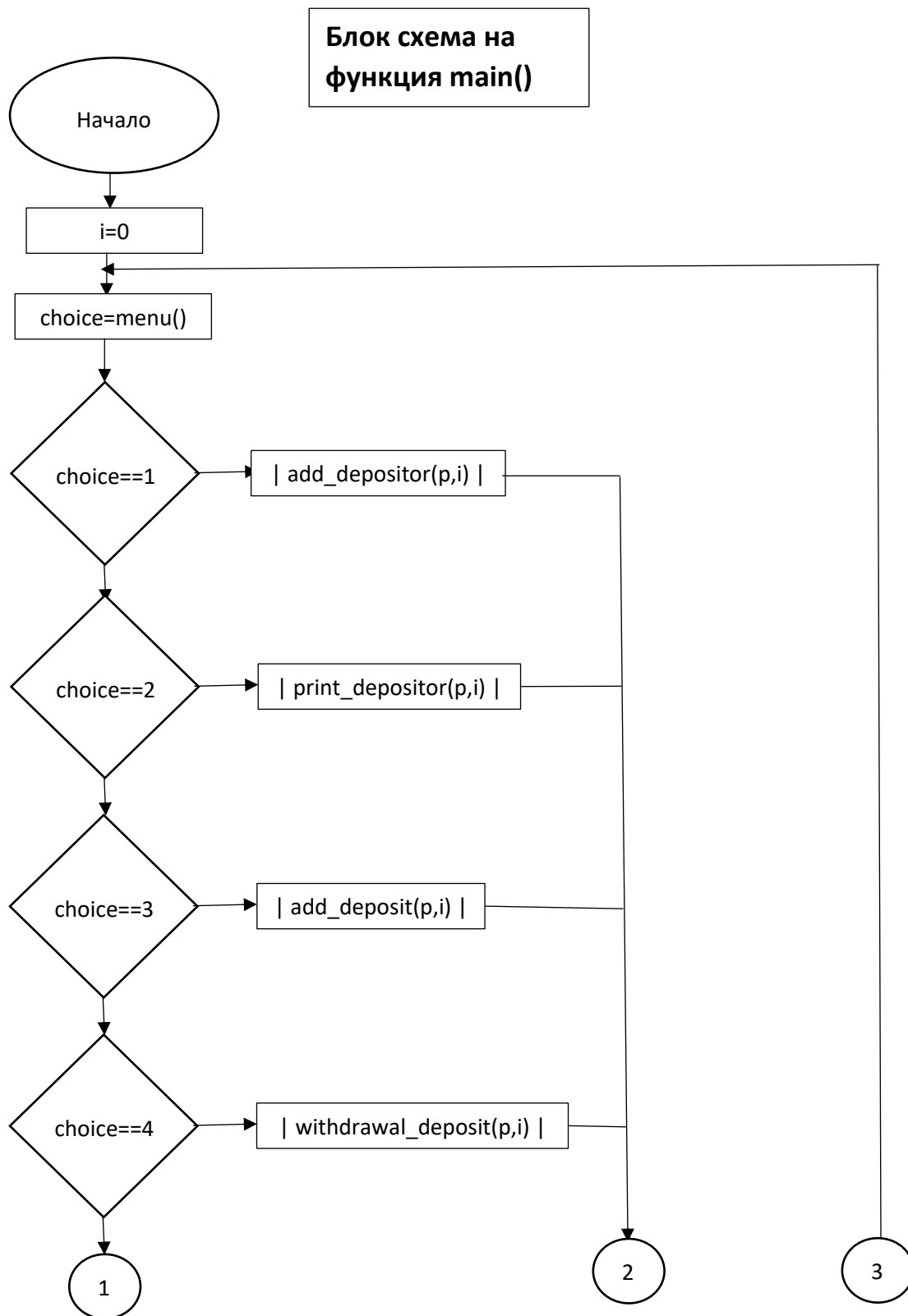
Стойността, която сме въвели се използва от мейна, където с помощта на switch в зависимост от 1-10 се извиква съответната функция. Ако напишем 1 ще се изведе функцията за добавяне на вложител, която служи за реализиране на В подточка.

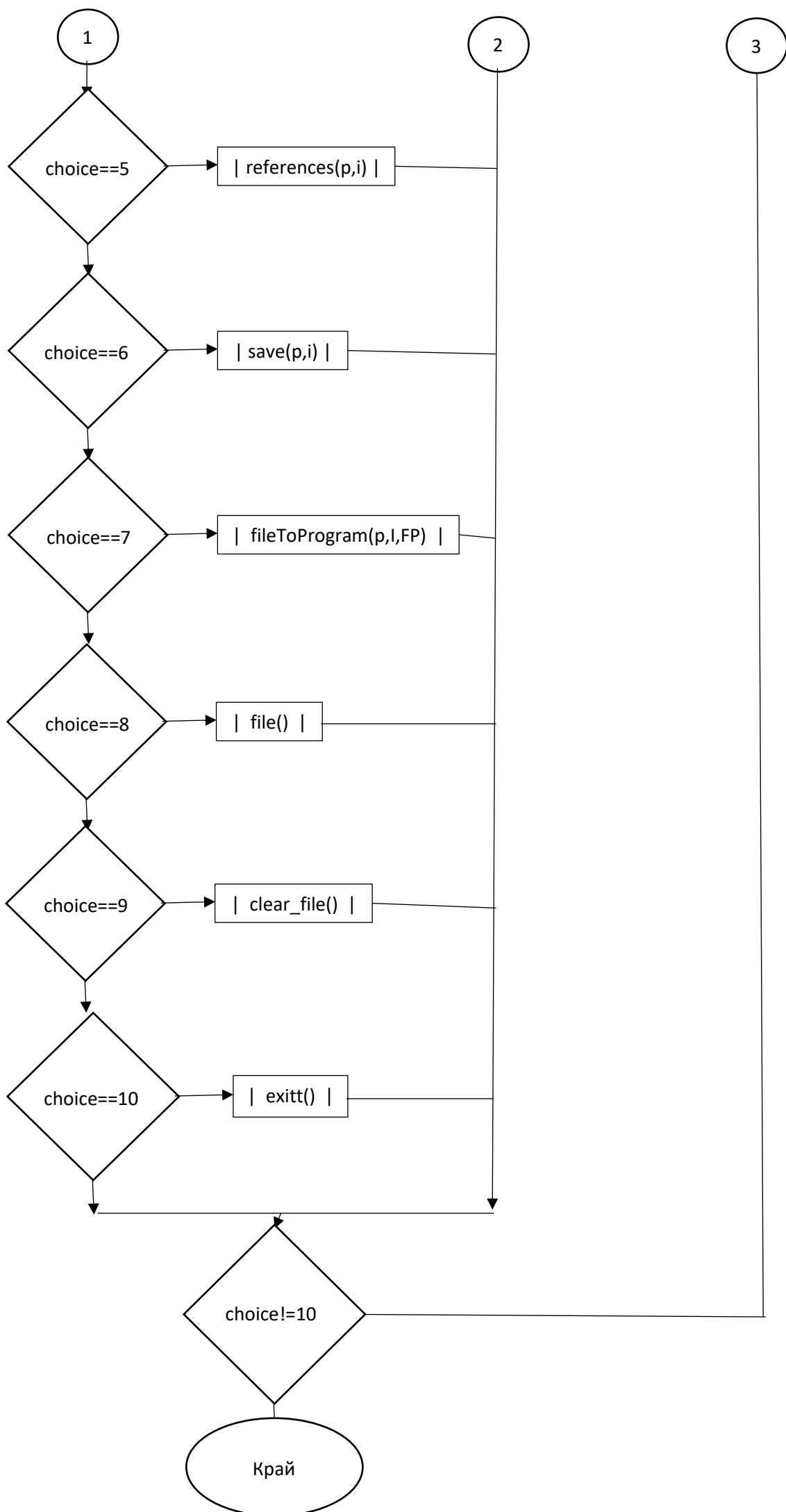
```
МЕНЮ
1. Добавяне на един вложител
2. Добавяне на много вложители
Въведете избор [1-2]:_
```

```
int main()
{
    person p[50];
    setlocale(LC_ALL, "Bulgarian");
    SetConsoleOutputCP(1251);
    SetConsoleCP(1251);

    int i=0, FP=0;
    int choice;
    do
    {
        // С помощта на цикъл повтаряме менюто, докато не се въведе стойност 10 (край).
        choice=menu();
        switch (choice)
        {
            case 1: add_depositor(p,i); break;
            case 2: print_depositor(p,i); break;
            case 3: add_deposit(p,i); break;
            case 4: withdrawal_deposit(p,i); break;
            case 5: references(p,i); break;
            case 6: save(p,i); break;
            case 7: fileToProgram(p,i,FP); break;
            case 8: file(); break;
            case 9: clear_file(); break;
            case 10: exit(0);
        }
    }while(choice!=10);
    system("pause");
    return 0;
}
```

В main-а се декларира масива Р с максимална бройка от 50 символа. Той служи за съхраняване на информацията за всеки един вложител. Използван е do-while цикъл, който се изпълнява до въвеждане на числото 10, което служи за изход от менюто и прекратяване на програмата. В тялото на цикъла се извиква функцията menu(), която връща стойност от 1 до 10. Тази стойност се проверява в switch, като за съответната стойност се извършва определена функция.



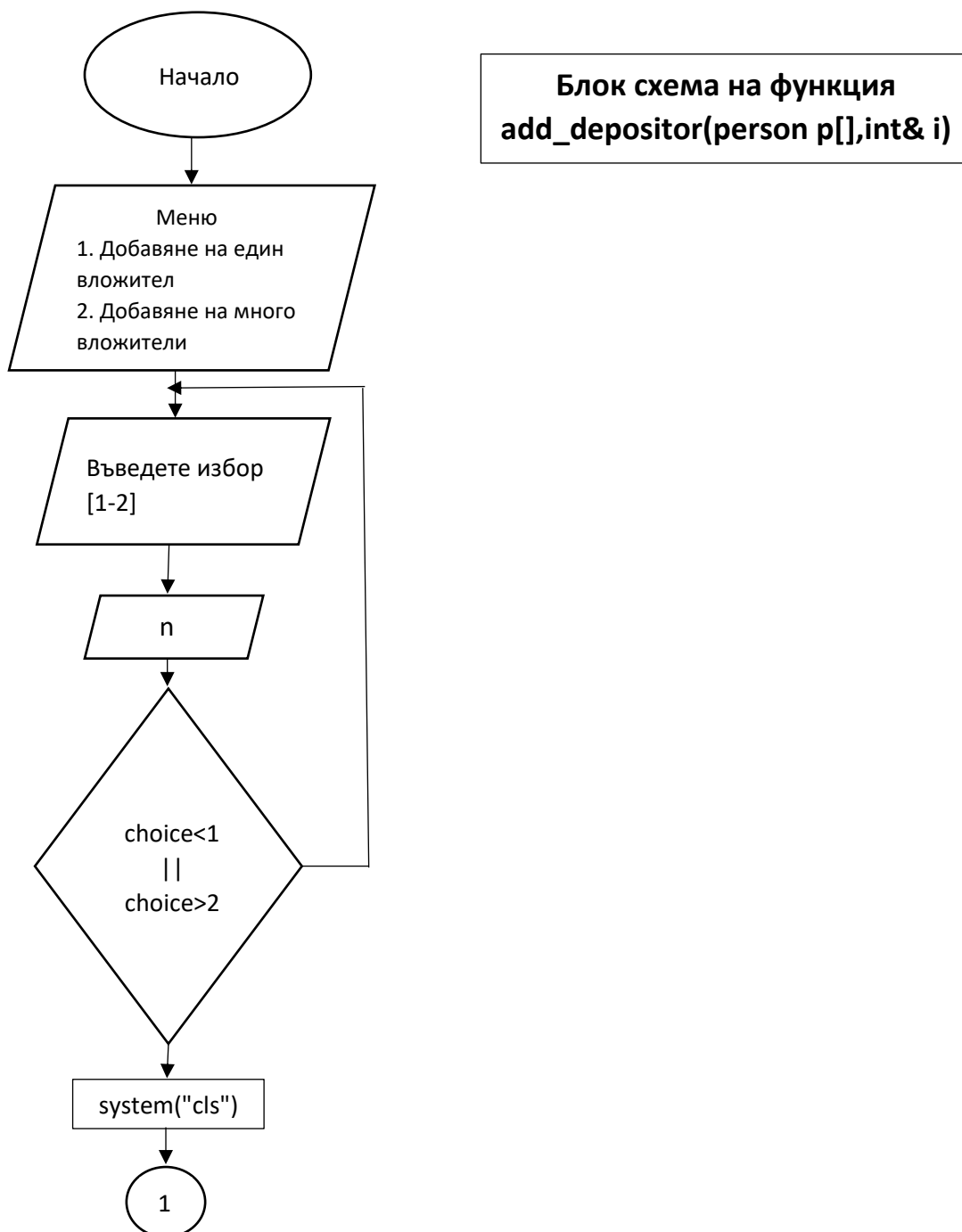


В) Добавяне на вложители:

а. Добавяне на нов вложител с един влог. Избира се валутата на влога (BGN (лев), USD (долар), EURO (евро));

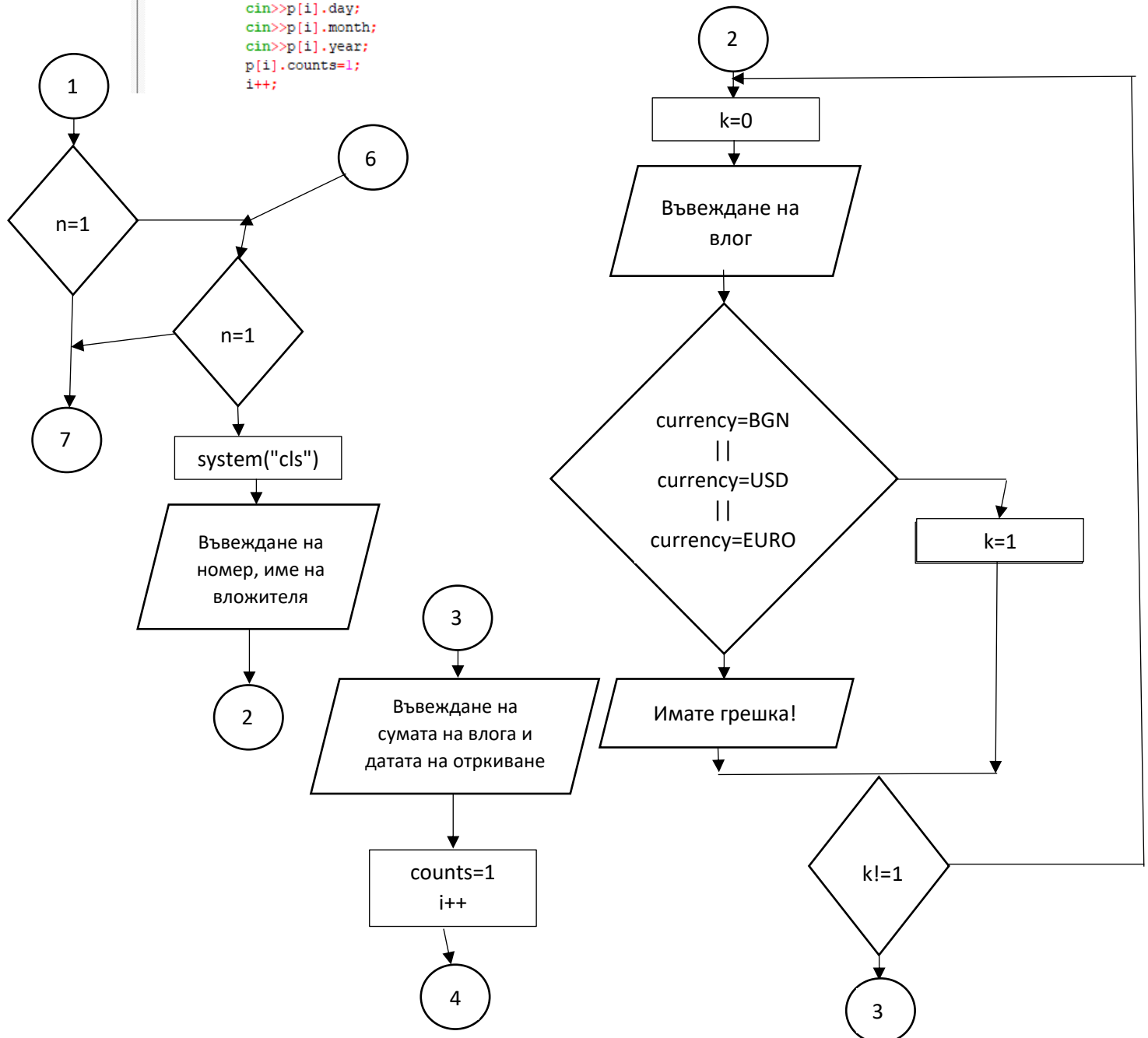
За изпълняването на тази част от задачата създавам функцията `add_depositor(p,i)`, която е от тип `void`, тъй като само извършва действия. Функцията използва входни данни за масива `p`, в който се запазва информацията за въведените вложители, а променливата `i` служи за брояч на въведените записи в масива. Функцията е съставена от първоначално от меню в което се избира дали да се добавя 1 вложител или много вложители на веднъж. После в зависимост от избора с помощта на `if` се изпълнява съответния избор.

```
void add_depositor(person p[],int& i)
{
    int n;
    cout<<"\n\t МЕНЮ";
    cout<<"\n 1. Добавяне на един вложител";
    cout<<"\n 2. Добавяне на много вложители";
    do{ // С помощта на цикъл взвеждаме нашия избор, който задължително трябва да е от 1-2
        // ако не е в този диапазон трябва потребителя да въведе нова стойност.
        cout<<"\n Въведете избор [1-2]:";
        cin>>n;
    }while(n<1 || n>2);
    system("cls"); // Изчистване на буфера след взведен правилен избор.
```



Ако изберем опция 1 ще се изпълни if и ще се изведат съобщения за попълване на данните на вложителя. Като в последователен ред ще се въведе: номер на карта; три имена; типа на влога, който ще се създава, а той трябва задължително да е (BGN/USD/EURO); сумата, която ще вкарва; и дата на създаване на влога. Тези данни се запазват в структура, която се запазва в масив в подреден номер.

```
if (n==1)
{ // Когато изберем първата опция системата запазва 1 вложител.
  while(n==1)
  { // С помощта на while цикъл е изпълнено първото попълнение за банковски данни.
    // След като се въведе 1 вложител се пита дали потребителят желае да въведе нов вложител.
    system("cls");
    cout<<"\tВъведете вложител:"<<endl;
    cout<<" Номер: ";
    cin>>p[i].card_number;
    cout<<" Три имена: ";
    cin.ignore();
    getline(cin,p[i].name);
    int k=0;
    do{ // С помощта на цикъл въвеждаме валутата, която задължително трябва да е BGN/USD/EURO.
      // ако сме въвели грешно трябва потребителят да въведе нова валута.
      cout<<" Изберете валута (BGN (лева), USD (долара), EURO (евро)): ";
      cin>>p[i].dp[0].currency;
      if(p[i].dp[0].currency=="BGN" || p[i].dp[0].currency=="USD" || p[i].dp[0].currency=="EURO")
        k=1;
      else
        cout<<" Имате грешка!"<<endl;
    }while(k!=1);
    cout<<" Сума: ";
    cin>>p[i].dp[0].amount;
    cout<<" Въведете дата на откриване на влога (ден месец година) ";
    cin>>p[i].day;
    cin>>p[i].month;
    cin>>p[i].year;
    p[i].counts=1;
    i++;
  }
}
```



```

struct deposit
{
    string currency;
    unsigned int amount;
};

struct person
{
    unsigned int card_number;
    string name;
    struct deposit dp[3];
    string day;
    string month;
    string year;
    int counts;
};

```

Структурата person е съставена от номера на картата от тип int, трите имена на вложителя от тип string, извиква се структурата deposit където в маси от 3 се съхранява съответния влог и неговата стойност, по отделно се съхраняват деня, месеца и годината от тип string, и накрая се въвежда и броя на създадените влогове до момента от тип int, благодарение на което въртим масива на структурата deposit.

Н) Допълнителни условия:

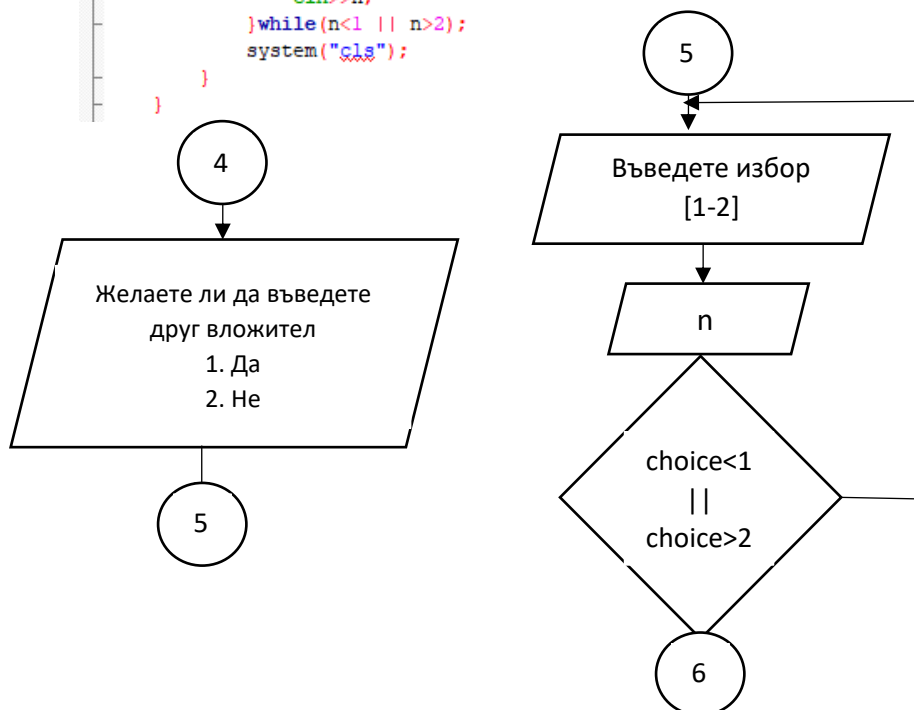
а. За точка В при добавянето на един вложител да се даде възможност, да се избере дали да се въведе нов или да се прекъсне въвеждането.

Добълнителното условие е изпълнено чрез малко подменю, което е вкарано в горния if, то се активира след въвеждането на данните на първия вложител. Ако потребителя желае да въведе нов вложител избира 1, а ако не избира 2. Ако е избрал 1 цикъла се завърта отново.

```

// Менюто което пита за добавяне на нов вложител.
cout<<"\n\t Желаете ли да въведете друг вложител?";
cout<<"\n 1. Да";
cout<<"\n 2. Не";
do{ // С помощта на цикъл въвеждаме нашия избор, който задължително трябва да е от 1-2
    // ако не е в този диапазон трябва потребителя да въведе нова стойност.
    cout<<"\n Въведете избор [1-2]:";
    cin>>n;
}while(n<1 || n>2);
system("cls");
}

```



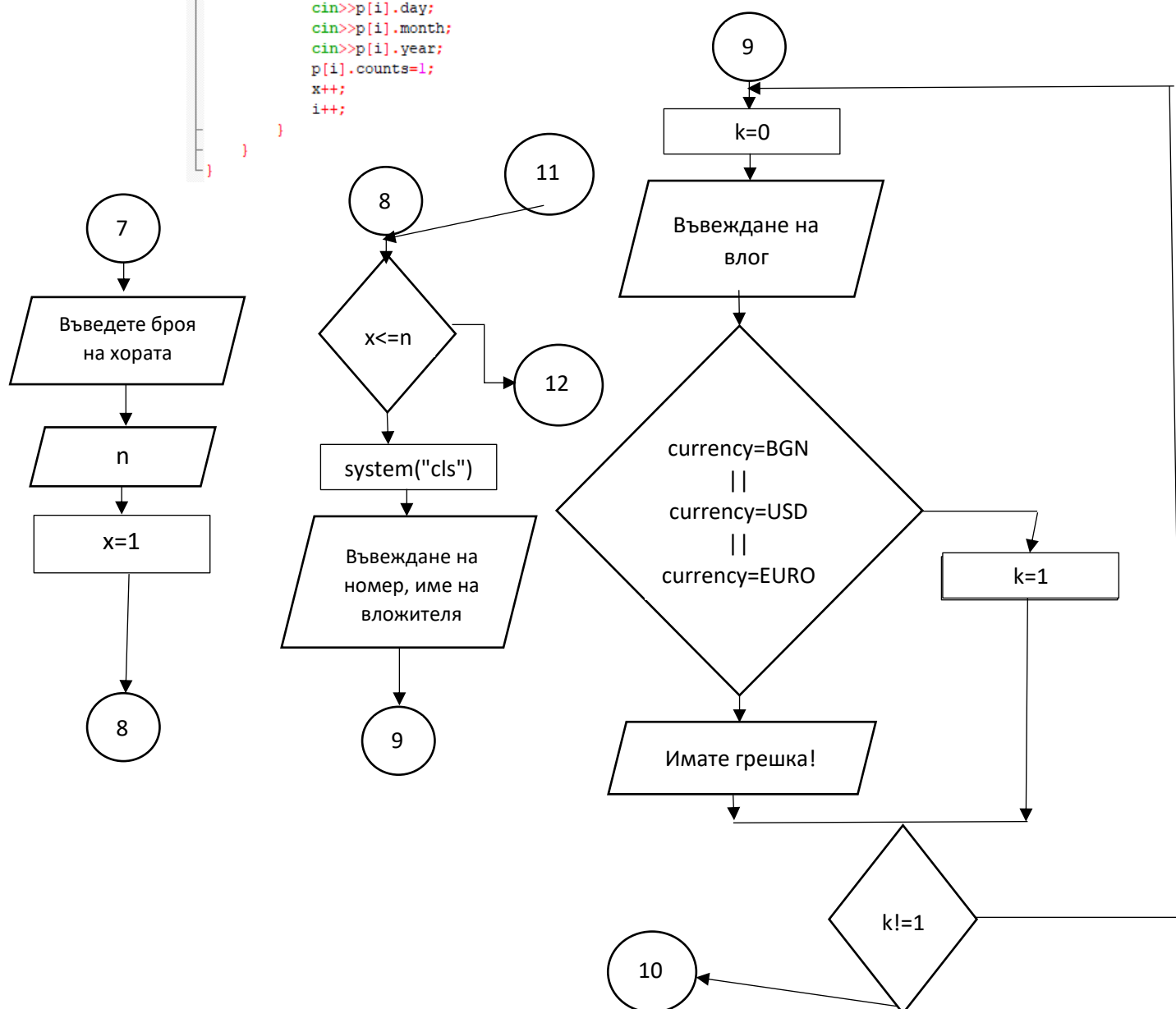
б. Добавяне на списък от вложители. Въвежда се цяло число n и след него n на брой вложители с един влог. Избира се валутата на влога (BGN (лев), USD (долар), EURO (евро)).

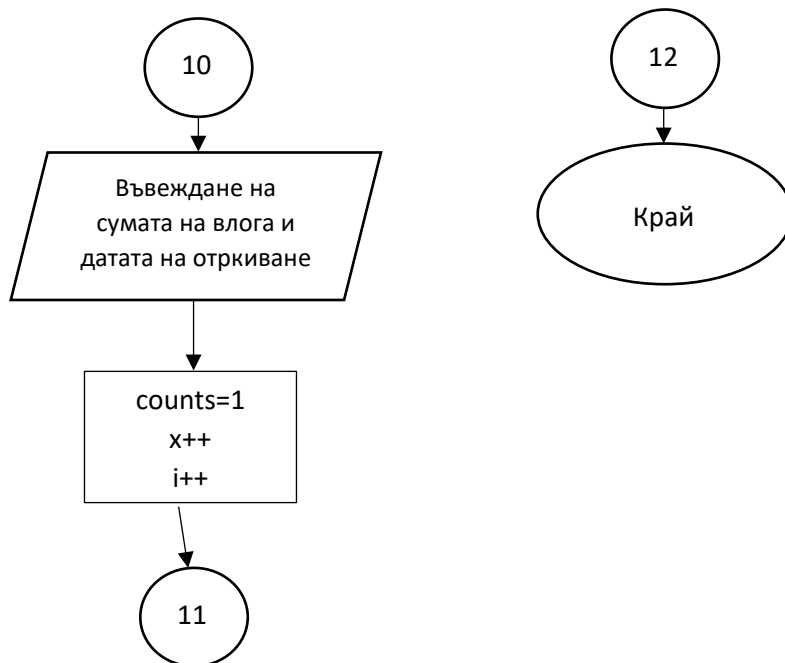
Ако сме избрали от главното менюто опция 2, за добавяне на много вложители, тогава n приема стойност 2 и се изпълнява else. В него първоначално се въвежда броя на вложителите, на които ще се създава влог. В последствие се върти цикъл, докато не се въведът всички хора, информацията която се въвежда е същата като при въвеждането на 1 вложител и отново се запазва в масива p под определен номер.

```

else
{
    cout<<" Въведете броя на хората: ";
    cin>>n;
    int x=1;
    while (x<=n)
    {
        system("cls");
        cout<<"\tВъведете вложител №"<<x<<":"<<endl;
        cout<<" Номер: ";
        cin>>p[i].card_number;
        cout<<" Тук имена: ";
        cin.ignore();
        getline(cin,p[i].name);
        int k=0;
        do{ // С помощта на цикъл въвеждаме валутата, която действително трябва да е BGN/USD/EURO.
            // ако сме въвели правилно трябва потребителя да въведе нова валута.
            cout<<" Изберете валута(BGN (лева), USD (долара),EURO (евро)): ";
            cin>>p[i].dp[0].currency;
            if (p[i].dp[0].currency=="BGN" || p[i].dp[0].currency=="USD" || p[i].dp[0].currency=="EURO")
                k=1;
            else
                cout<<" Имате грешка!"<<endl;
        }while(k!=1);
        cout<<" Сума: ";
        cin>>p[i].dp[0].amount;
        cout<<" Въведете дата на откриване на влога. (ден месец година) ";
        cin>>p[i].day;
        cin>>p[i].month;
        cin>>p[i].year;
        p[i].counts=1;
        x++;
        i++;
    }
}

```





Упътване за употреба на подточка В

След като сме селектирали от главното меню опцията за добавяне на вложител, излиза ново меню на което трябва да изберем дали ще въвеждаме един или много вложители. Както се вижда на конзолата, при въвеждане на число различно от 1 или 2 се пита за ново въвеждане и така докато не се въведе число в този диапазон.

```

МЕНЮ
1. Добавяне на един вложител
2. Добавяне на много вложители
Въведете избор [1-2]:3
Въведете избор [1-2]:1
  
```

Ако въведем 1 ще се активират съобщенията за добавяне на един вложител.

Един след друг се въвеждат: номер на карта; три имена вида на спестовния влог, който трябва да е (BGN/USD/EURO), ако не е се извежда ново питане за въвеждане и така докато не се въведе правилно; дата на откриване на влога.

След преключване с въвеждането на коректни данни, излиза менюто което пита дали потребителя желае да въведе нов вложител. Ако се натисне 1 отново ще се покажат полетата за въвеждане на вложител, ако обаче натиснем 2 ще се върнем на главното меню.

```

Въведете вложител:
Номер: 526574
Три имена: Стоян Петров Генчев
Изберете влог(BGN (лева), USD (долара),EURO (евро)): БГН
Имате грешка!
Изберете влог(BGN (лева), USD (долара),EURO (евро)): BGN
Сума: 2140
Въведете дата на откриване на влога.(ден/месец/година) 21/05/2018

Желаете ли да въведете друг вложител?
1. Да
2. Не
Въведете избор [1-2]:2
  
```

Ако обаче от менюто изберем 2 за въвеждане на много вложители, тогава ще ни излезе запитване колко хора желаем да въведем.

```

МЕНЮ
1. Добавяне на един вложител
2. Добавяне на много вложители
Въведете избор [1-2]:2
  
```

В случая избираме да въведем 2 вложителя

```

Въведете броя на хората: 2
  
```

Отново въвеждаме същата информация както и при добавяне на един вложител. След въвеждането на 2та вложители програмата автоматично ни връща на главното меню.

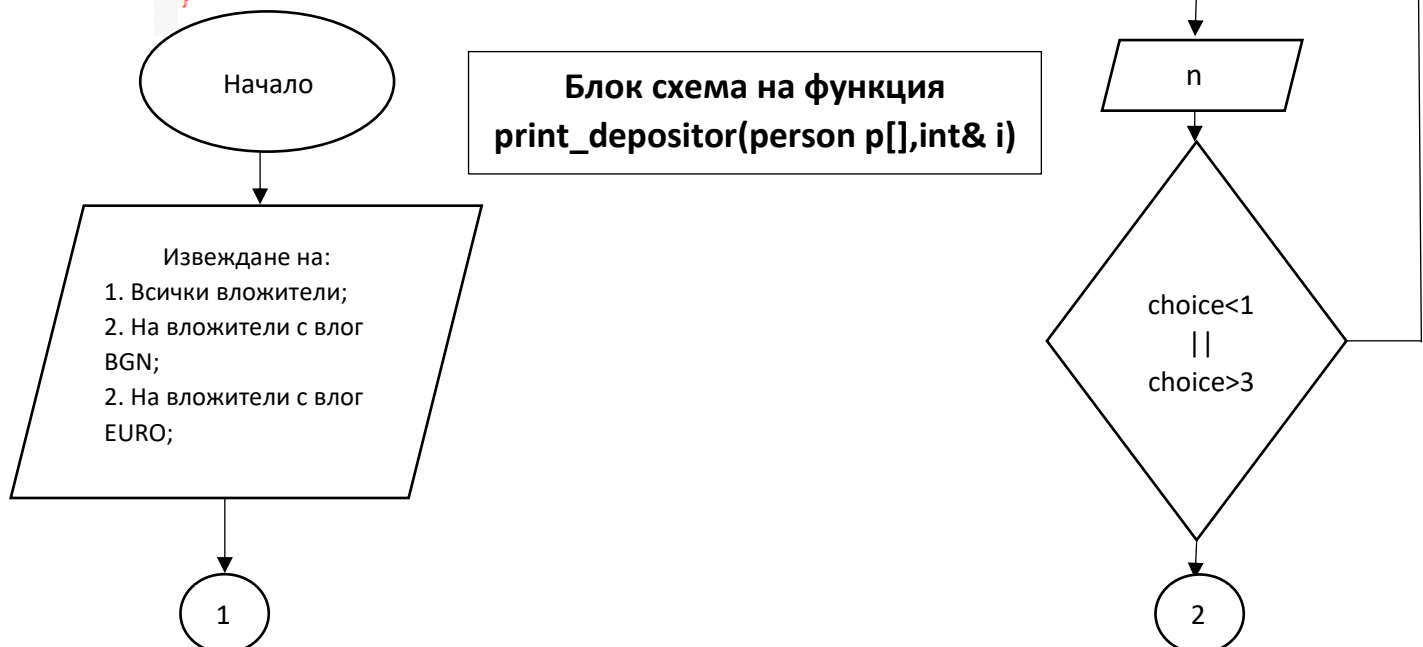
```
Въведете вложител №1:
Номер: 241531
Три имена: Иван Сотиров Тодоров
Изберете влог(BGN (лева), USD (долара),EURO (евро)): BGN
Сума: 2415
Въведете дата на откриване на влога.(ден/месец/година) 15/11/2020
```

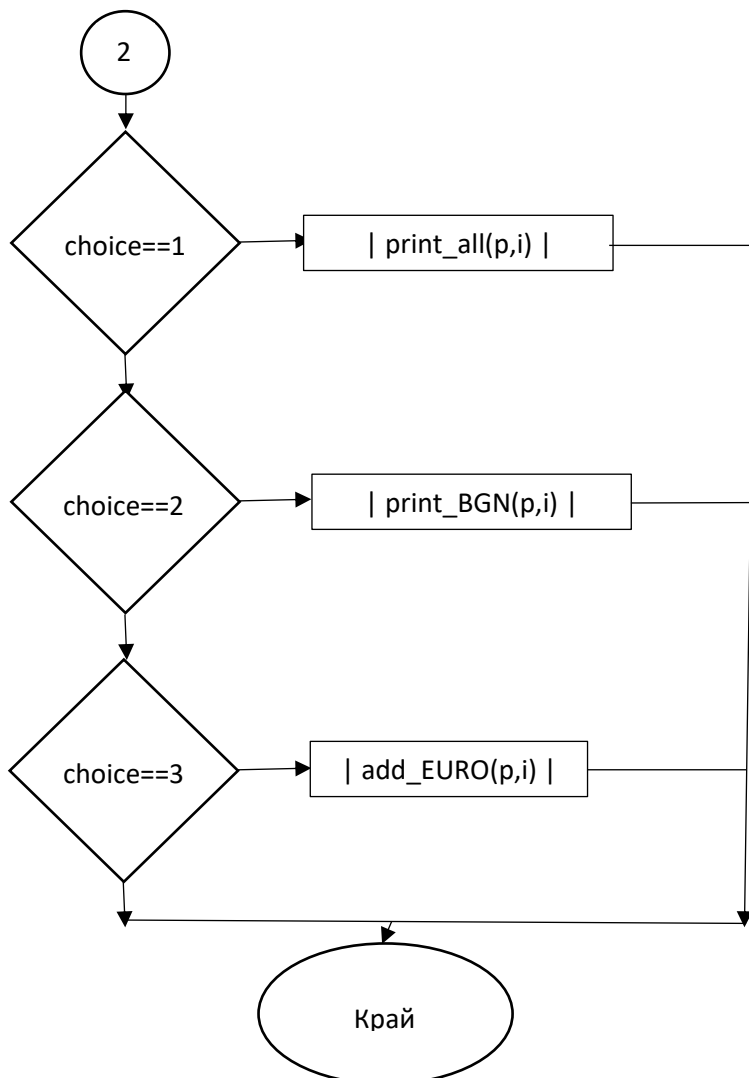
```
Въведете вложител №2:
Номер: 241521
Три имена: Мария Колева Романова
Изберете влог(BGN (лева), USD (долара),EURO (евро)): EURO
Сума: 3520
Въведете дата на откриване на влога.(ден/месец/година) 25/02/2019
```

С. Извеждане на всички вложители на екрана:

За изпълняването на тази част от задачата създавам функцията `print_depositor(p[],i)`, която е от тип `void`, тъй като само извършва действия. Функцията използва входни данни за масива `p`, в който е запазена информацията за въведените вложители, а променливата `i` показва броя на въведените записи в масива. Функцията е съставена от първоначално от меню в което се избира дали да се изведат всички вложители, или само тези с влог BGN, или тези с влог EURO. После в зависимост от избора с помощта на `switch` се изпълнява съответния избор.

```
void print_depositor(person p[],int i)
{
    int n;
    cout<<"\n Извеждане на:";
    cout<<"\n 1. Всички вложители:";
    cout<<"\n 2. На вложители с влог BGN:";
    cout<<"\n 2. На вложители с влог EURO.";
    do{ // С помощта на цикъл извеждаме нашия избор, който задължително трябва да е от 1-3
        // ако не е в този диапазон трябва потребителя да въведе нова стойност.
        cout<<"\n Въведете избор [1-3]:";
        cin>>n;
    }while(n<1 || n>3);
    system("cls");
    switch(n)
    {
        case 1: print_all(p,i); break;
        case 2: print_BGN(p,i); break;
        case 3: print_EURO(p,i);
    }
}
```



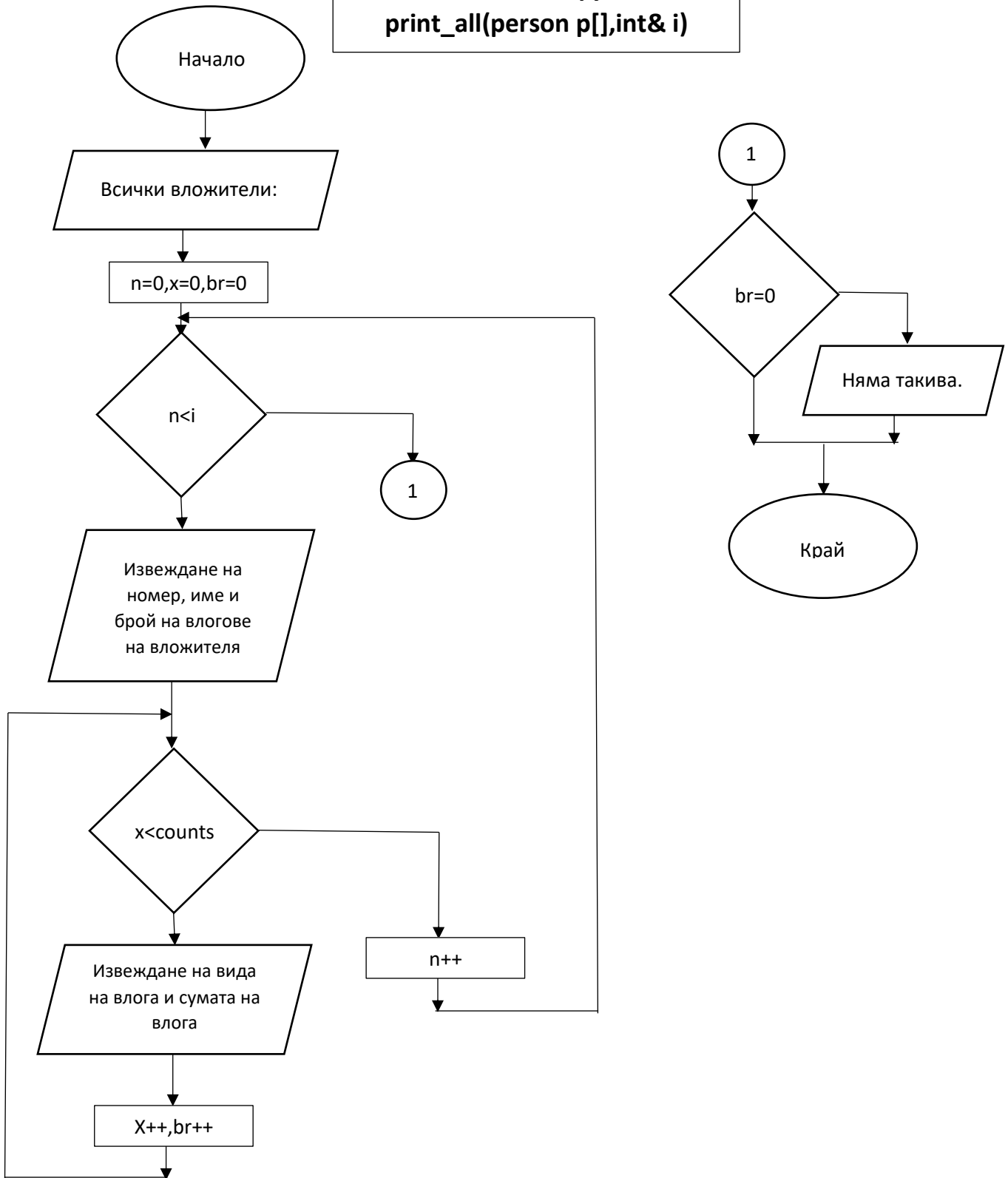


Функцията `print_all(p[],i)`, която е от тип `void`, изкарва на екрана всички записи в програмата. Функцията използва входни данни за масива `p`, в който е запазена информацията за въведените вложители, а променливата `i` показва броя на въведените записи в масива. Цялата информация се изкарва на екрана с помощта на цикъл, който върти главния масив `p`, а чрез под цикъл се върти броят на всички влогове, които един вложител има. Накрая чрез `if` се проверява дали брояча `br` се е увеличил, ако не следователно няма записи в програмата.

```

void print_all(person p[],int i) // Подфункция за функция print_depositor().
{
    int br=0;
    cout<<"\t Всички вложители:"<<endl;
    for (int n=0; n<i; n++)
    { // Цикъл, който върти елементите на масива.
        int x=0;
        cout<<" Номер: ";
        cout<<p[n].card_number<<endl;
        cout<<" Три имена: ";
        cout<<p[n].name<<endl;
        cout<<" Брой влогове: "<<p[n].counts<<endl<<endl;
        while(x<p[n].counts)
        { // Цикъл, който върти елементите на подмасива.
            cout<<" Влог №"<<x+1<<": ";
            cout<<p[n].dp[x].currency<<endl;
            cout<<" Сума: ";
            cout<<p[n].dp[x].amount<<endl<<endl;
            br++;
            x++;
        }
        cout<<endl;
    }
    if (br==0)
        cout<<" Няма такива."<<endl<<endl;
}
  
```

**Блок схема на функция
print_all(person p[],int& i)**



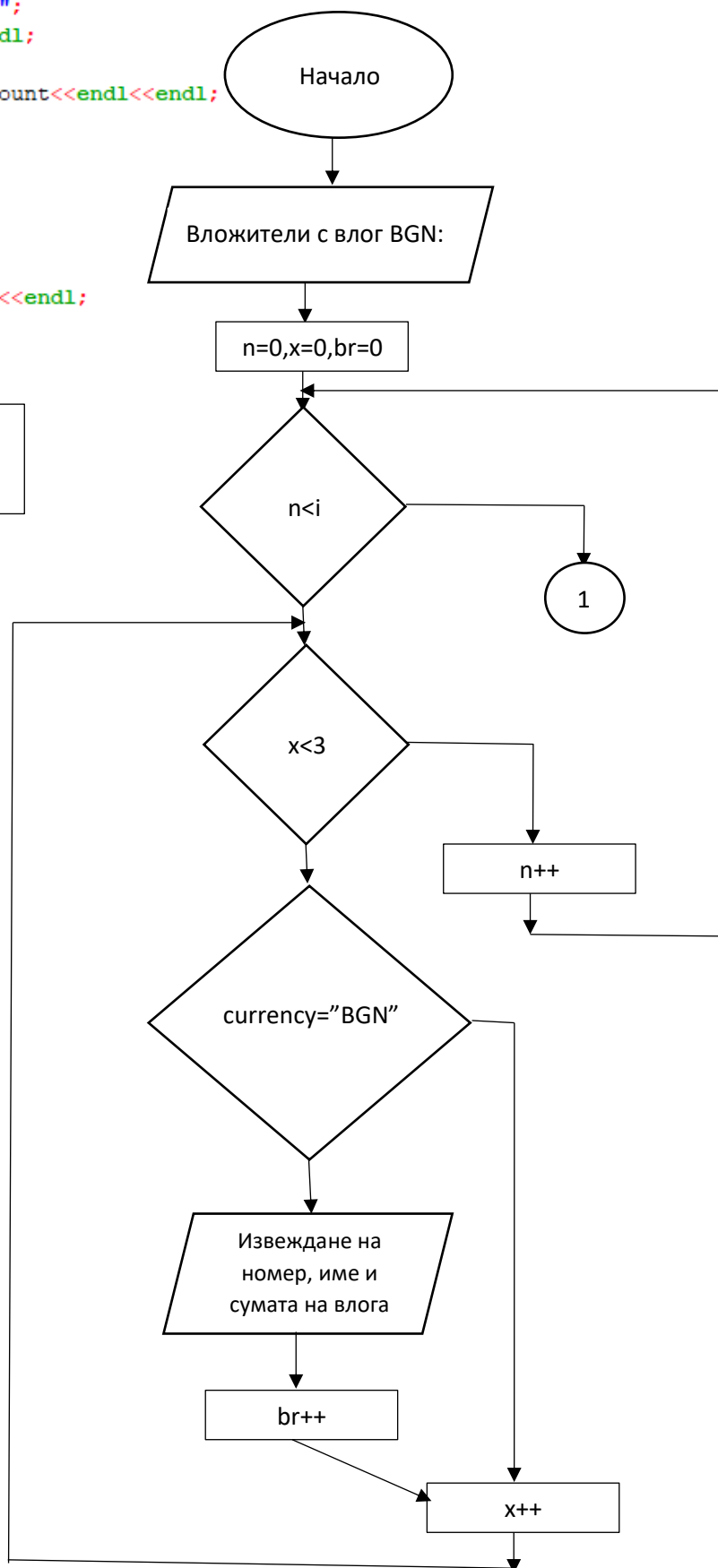
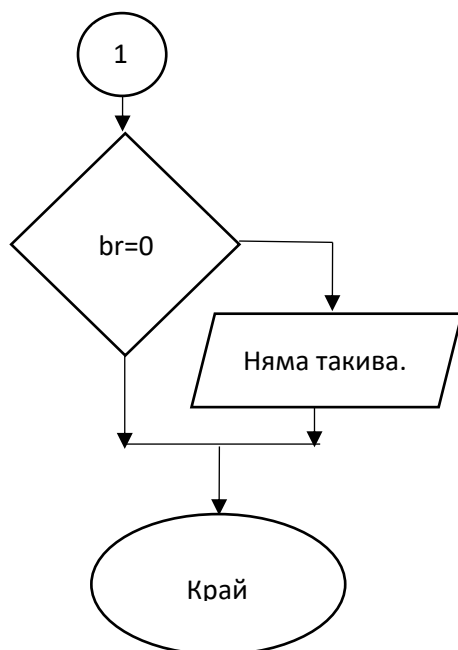
Функцията print_BGN(p[],i), която е от тип void, извежда на екрана всички хора с влог BGN. Функцията използва входни данни за масива p, в който е запазена информацията за въведените вложители, а променливата i показва броя на въведените записи в масива. Цялата информация се преглежда с помощта на цикъл, който върти главния масив p, а чрез под цикъл се върти броят на всички влогове, които един вложител има. Чрез if се проверява дали съответния влог е от тип BGN, ако е го изкарва на екрана. Накрая чрез if се проверява дали брояча br се е увеличил, ако не следователно няма записи в програмата.

```

void print_BGN(person p[],int i) // Подфункция за функция print_depositor().
{
    int br=0;
    cout<<"\tВложители с влог BGN:"<<endl;
    for (int n=0; n<i; n++)
    {
        // Цикъл, който върти елементите на масива.
        int x=0;
        while(x<3)
        {
            // Цикъл, който върти елементите на подмасива.
            if(p[n].dp[x].currency=="BGN")
            {
                // Изкарват се хората с влог BGN.
                cout<<" Номер: ";
                cout<<p[n].card_number<<endl;
                cout<<" Три имена: ";
                cout<<p[n].name<<endl;
                cout<<" Сума: ";
                cout<<p[n].dp[x].amount<<endl<<endl;
                br++;
            }
            x++;
        }
    }
    if (br==0)
        cout<<" Няма такива."<<endl<<endl;
}

```

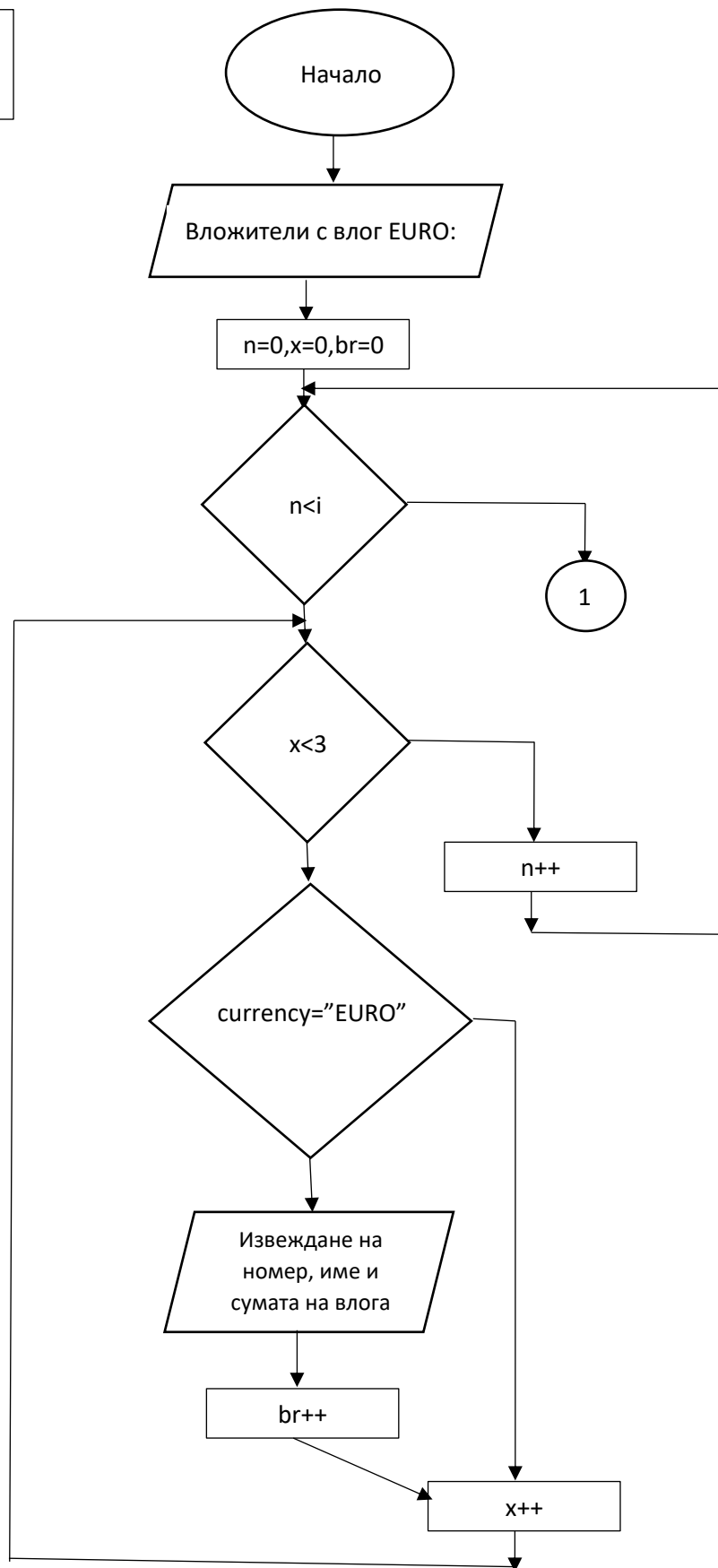
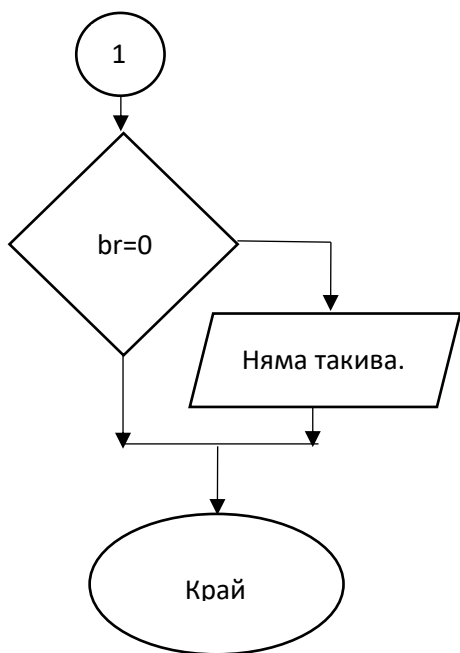
Блок схема на функция
print_BGN(person p[],int& i)



Функцията `print_EURO(p[],i)`, която е от тип `void`, изкарва на екрана всички хора с влог `EURO`. Функцията използва входни данни за масива `p`, в който е запазена информацията за въведените вложители, а променливата `i` показва броя на въведените записи в масива. Цялата информация се преглежда с помощта на цикъл, който върти главния масив `p`, а чрез под цикъл се върти броят на всички влогове, които един вложител има. Чрез `if` се проверява дали съответния влог е от тип `EURO`, ако е го изкарва на екрана. Накрая чрез `if` се проверява дали брояча `br` се е увеличил, ако не следователно няма записи в програмата.

```
void print_EURO(person p[],int i) // Подфункция за функция print_depositor().
{
    cout<<endl<<"\tВложители с влог EURO:"<<endl;
    int br=0;
    for (int n=0; n<i; n++)
    { // Цикъл, който върти елементите на масива.
        int x=0;
        while(x<3)
        { // Цикъл, който върти елементите на подмасива.
            if(p[n].dp[x].currency=="EURO")
            { // Изкарват се хората с влог EURO.
                cout<<" Номер: ";
                cout<<p[n].card_number<<endl;
                cout<<" Три имена: ";
                cout<<p[n].name<<endl;
                cout<<" Сума: ";
                cout<<p[n].dp[x].amount<<endl<<endl;
                br++;
            }
            x++;
        }
    }
    if (br==0)
        cout<<" Няма такива."<<endl<<endl;
}
```

**Блок схема на функция
print_EURO(person p[],int& i)**



Упътване за употреба на подточка С

След като сме селектирали от главното меню опцията за извеждане на вложители, излиза ново меню на което трябва да изберем дали да се покажат всички вложители, дали тези, които имат влог в BGN, или тези, които имат влог в EURO. Както се вижда на конзолата, при въвеждане на число различно от 1, 2 или 3 се пита за ново въвеждане и така докато не се въведе число в този диапазон.

```
Извеждане на:
1. Всички вложители;
2. На вложители с влог BGN;
3. На вложители с влог EURO.
Въведете избор [1-3]:4
Въведете избор [1-3]:1
```

Ако въведем 1 ще се покажат всички вложители. Ако обаче нямаме въведени данни в системата, ще се въведе съобщението, че няма такива.

```
Всички вложители:
Няма такива.
```

Ако използваме въведените данни от точка В и натиснем от менюто 1 те ще се изведат.

```
Всички вложители:
*****
Номер: 526574
Три имена: Стоян Петров Генчев
Брой влогове: 1

Влог №1: BGN
Сума: 2140
*****
Номер: 241531
Три имена: Иван Сотиров Тодоров
Брой влогове: 1

Влог №1: BGN
Сума: 2415
*****
Номер: 241521
Три имена: Мария Колева Романова
Брой влогове: 1

Влог №1: EURO
Сума: 3520
*****
```

Ако изберем от подменюто да се изведат хората, които имат само влогове от BGN, то те ще се изведът.

```
Вложители с влог BGN:
*****
Номер: 526574
Три имена: Стоян Петров Генчев
Сума: 2140
*****
Номер: 241531
Три имена: Иван Сотиров Тодоров
Сума: 2415
*****
```

Ако изберем от подменюто да се изведат хората, които имат само влогове от EURO, то те ще се изведът.

```
Вложители с влог EURO:
*****
Номер: 241521
Три имена: Мария Колева Романова
Сума: 3520
*****
```

2. Допълнение Първо

D. Добавяне на нов влог:

За изпълняването на тази част от задачата създавам функцията `add_deposit(p,i)`, която е от тип `void`, тъй като само извършва действия. Функцията използва входни данни за масива `p`, в който е запазена информацията за въведените вложители, а променливата `i` показва броя на записите в масива. Функцията е съставена от две части. Първоначално се въвежда номера на картата на търсения вложител. Чрез помощта на цикъл се извършва проверка в масива. Ако не съществува такъв човек, се изписва съответното съобщение.

```
void add_deposit(person p[],int i)
{
    int number;
    cout<<"\tДобавяне на нов влог"<<endl;
    cout<<" Въведете номер на сметка: ";
    cin>>number;
    int pos=-1;
    for(int n=0; n<i; n++)
    {
        // Цикъл, който върти елементите на масива.
        if(p[n].card_number==number)
        {
            // Намиране на номера в системата.
            pos=n;
            break;
        }
    }
}
```

b. Ако не съществува вложител с този номер да се изведе подходящо съобщение;

```
if (pos==-1) // Ако номера не е намерен.
    cout<<"\n Не съществува такъв номер в системата!"<<endl;
else // Ако е намерен.
```

а. Всеки вложител (по номер на възложител) с по-малко от 3 влога може да му се създаде нов влог за валута по която няма влог с минимална стойност 500;

Ако обаче съществува такъв човек, неговия подреден номер в масива се запазва. После се проверява колко влога вече има създадени, ако има 1 му се предлагат съответните възможности.

```
else // Ако е намерен.
{
    if(p[pos].counts==1) // Ако има 1 влог.
    {
        if(p[pos].dp[0].currency=="BGN") // Ако е в BGN.
        {
            int k=0;
            do{ // Проверка за правилно въведена валута.
                cout<<" Изберете влог(USD (долара), EURO (евро)): ";
                cin>>p[pos].dp[1].currency;
                if(p[pos].dp[1].currency=="USD" || p[pos].dp[1].currency=="EURO")
                    k=1;
                else
                    cout<<" Имате грешка!"<<endl;
            }while(k!=1);
            add_amount(p,pos,1); // Изпълняване на подфункция.
        }
        else if(p[pos].dp[0].currency=="EURO") // Ако е в EURO.
        {
            int k=0;
            do{ // Проверка за правилно въведена валута.
                cout<<" Изберете влог(BGN (лева), USD (долара)): ";
                cin>>p[pos].dp[1].currency;
                if(p[pos].dp[1].currency=="BGN" || p[pos].dp[1].currency=="USD")
                    k=1;
                else
                    cout<<" Имате грешка!"<<endl;
            }while(k!=1);
            add_amount(p,pos,1); // Изпълняване на подфункция.
        }
        else // Ако е в USD.
        {
            int k=0;
            do{ // Проверка за правилно въведена валута.
                cout<<" Изберете влог(BGN (лева), EURO (евро)): ";
                cin>>p[pos].dp[1].currency;
                if(p[pos].dp[1].currency=="BGN" || p[pos].dp[1].currency=="EURO")
                    k=1;
                else
                    cout<<" Имате грешка!"<<endl;
            }while(k!=1);
            add_amount(p,pos,1); // Изпълняване на подфункция.
        }
    }
}
```

С помощта на под функция add_amount() се вкарва желаната сума, която задължително трябва да е по-голяма от 500.

```
void add_amount(person p[],int pos,int x) // Подфункция за функция add_deposit().
{
    cout<<" Въведете сума от минимална стойност 500: ";
    cin>>p[pos].dp[x].amount;
    while (p[pos].dp[x].amount<500)
    { // Проверка за правилно въведена сума.
        cout<<" Трябва по-голяма сума: ";
        cin>>p[pos].dp[x].amount;
    }
    p[pos].counts=x+1; // Броя на влоговете се вдига с 1.
}
```

Ако има вече 2 създадени влога му се предлага съответната последна възможност. И се изкарва съобщение, дали желае да му се създаде съответния влог.

```
else if(p[pos].counts==2) // Ако има 2 съществени влога.
{
    if((p[pos].dp[0].currency=="BGN"&&p[pos].dp[1].currency=="EURO") || (p[pos].dp[1].currency=="BGN"&&p[pos].dp[0].currency=="EURO")) // Ако е в BGN & EURO.
    {
        cout<<" Можете да имате само още един влог във валута USD (долара): "<<endl;
        string s;
        int k=0;
        do{ // Проверка за правилно въведен избор.
            cout<<" Желаете ли да ми бъде открит? (да/не) ";
            cin>>s;
            if(s=="да")
            {
                k=1;
                p[pos].dp[2].currency="USD";
                add_amount(p,pos,2); // Изпълняване на допълнителна.
            }
            else if(s=="не")
                break;
            else
                cout<<" Имате грешка!"<<endl;
        }while(k!=1);
    }
    if((p[pos].dp[0].currency=="BGN"&&p[pos].dp[1].currency=="USD") || (p[pos].dp[1].currency=="BGN"&&p[pos].dp[0].currency=="USD")) // Ако е в BGN & USD.
    {
        cout<<" Можете да имате само още един влог във валута EURO (долара): "<<endl;
        string s;
        int k=0;
        do{ // Проверка за правилно въведен избор.
            cout<<" Желаете ли да ми бъде открит? (да/не) ";
            cin>>s;
            if(s=="да")
            {
                k=1;
                p[pos].dp[2].currency="EURO";
                add_amount(p,pos,2); // Изпълняване на допълнителна.
            }
            else if(s=="не")
                break;
            else
                cout<<" Имате грешка!"<<endl;
        }while(k!=1);
    }

    if((p[pos].dp[0].currency=="EURO"&&p[pos].dp[1].currency=="USD") || (p[pos].dp[1].currency=="EURO"&&p[pos].dp[0].currency=="USD")) // Ако е в USD & EURO.
    {
        cout<<" Можете да имате само още един влог във валута BGN (долара): "<<endl;
        string s;
        int k=0;
        do{ // Проверка за правилно въведен избор.
            cout<<" Желаете ли да ми бъде открит? (да/не) ";
            cin>>s;
            if(s=="да")
            {
                k=1;
                p[pos].dp[2].currency="BGN";
                add_amount(p,pos,2); // Изпълняване на допълнителна.
            }
            else if(s=="не")
                break;
            else
                cout<<" Имате грешка!"<<endl;
        }while(k!=1);
    }
}
}
```

с. При опит за добавяне на влог на вложител с 3 влога се изписва подходящо съобщение;

Ако има 3 му се изкарва съобщение, че няма възможност на повече влогове. И с това свършва функцията.

```
else // Ако има 3 създадени влога.
    cout<<"\n Не могат да бъдат открити нови влогове, тъй като максималния брой е достигнат!"<<endl;
}
```

Упътване за употреба на подточка D

След като сме селектирали от главното меню опцията за добавяне на нов влог, излиза запитване за търсения номер на сметка. Ако въведем номер, който не съществува се изкарва съответното съобщение, а програмата се връща на главното меню.

```
Добавяне на нов влог
Въведете номер на сметка: 214521

Не съществува такъв номер в системата!

МЕНЮ
1. Добавяне на вложител
2. Извеждане на всички вложители
3. Добавяне на нов влог
4. Изтегляне на влог
5. Справки
6. Запазване на въведените данни във файл
7. Прехвърляне на данните от файла в програмата
8. Извеждане на данните от файла
9. Изчистване на запазените данни във файла
10. Край

Въведете избор [1-10]:
```

Ако въведем номер, който съществува в системата се изкарва съобщение с възможностите за откриване на нов влог. Ако се въведе грешен влог от посочените се изкарва грешка и се пита за ново въвеждане.

В случая въвеждаме номер, който съществува в системата с един влог в BGN. За това изборите, които стоят пред него са само влогове в евро и долара. Ако се опитаме да създадем влог в BGN, системата дава грешка. Когато изберем влога ни излиза запитване за желаната сума, която смятаме да вкараме, тя обаче трябва да е повече от 500. Както се вижда на снимката, при въвеждане на по ниска сума, се изкарва съобщение. След правилно въведени данни програмата се връща на главното меню.

```
Добавяне на нов влог
Въведете номер на сметка: 526574
Изберете влог(USD (долара), EURO (евро)): BGN
Имате грешка!
Изберете влог(USD (долара), EURO (евро)): EURO
Въведете сума от минимална стойност 500: 200
Трябва по-голяма сума: 550
```

Ако въведем номер, който съществува в системата с два влога, се изкарва съответното съобщение. При избиране, че желаем да създадем такъв влог се въвежда сума, която трябва да е по-голяма от 500.

```
Добавяне на нов влог
Въведете номер на сметка: 526574
Можете да имате само още един влог във валута USD (долара):
Желаєте ли да ви бъде открит? (да/не) може
Имате грешка!
Желаєте ли да ви бъде открит? (да/не) да
Въведете сума от минимална стойност 500: 250
Трябва по-голяма сума: 670
```

За да сме сигурни, че се е създавал такъв влог, от главното меню викаме опцията да се изведат всички вложители.

```
Всички вложители:
Номер: 526574
Три имена: Стоян Петров Генчев
Брой влогове: 3

Влог №1: BGN
Сума: 2140

Влог №2: EURO
Сума: 550

Влог №3: USD
Сума: 670
```

Ако се опитаме да създадем нов влог на Стоян ще се изведе следното съобщение.

```
Добавяне на нов влог
Въведете номер на сметка: 526574

Не могат да бъдат открити нови влогове, тъй като максималния брой е достигнат!
```

3. Допълнение Второ

Е. Изтегляне на влог

За изпълняването на тази част от задачата създавам функцията `withdrawal_deposit(p,i)`, която е от тип `void`, тъй като само извършва действия. Функцията използва входни данни за масива `p`, в който е запазена информацията за въведените вложители, а променливата `i` показва броя на записите в масива. Функцията е съставена от първоначално от меню, в което се избира дали да се добавя тегли сума от влог, или да се закрие профила на вложител. После в зависимост от избора с помощта на `if` се изпълнява съответния избор.

```
void withdrawal_deposit(person p[],int& i)
{
    int n, number, money;
    cout<<"\n\t МЕНЮ";
    cout<<"\n 1. Теглене на сума от влог";
    cout<<"\n 2. Прекратяване на профила";
    do{ // С помощта на цикъл въвеждаме нашия избор, който задължително трябва да е от 1-2
        // ако не е в този диапазон трябва потребителя да въведе нова стойност.
        cout<<"\n Въведете избор [1-2]:";
        cin>>n;
    }while(n<1 || n>2);
    system("cls");
}
```

а. Въвеждат се номер на вложител, избира се влога от който ще се тегли и се изтегля цялата или част от сумата по влога.

Ако изберем от менюто 1, ще се изпълни следната част на `if`. Въвежда се номера на карта на вложителя, като ако не е намерен такъв човек в системата се изкарва грешка.


```

int br=0;
if (n==1) // Ако сме избрали опция 1.
{
    cout<<" Въведете номера на вложителя: ";
    cin>>number;
    while(br==0)
    { // Цикъл, който върти елементите на масива.
        for (int x=0; x<i; x++)
        {
            if(p[x].card_number==number)
            { // Намиране на номера в системата.
                cout<<endl;
                deposits(p,x); // Изпълняване на подфункция.
                br++;
            }
        }
        if (br==0) // Ако не е намерен в системата.
        {
            if(i==0) // Ако базата данни е празна.
            {
                cout<<" Базата данни е празна."<<endl;
                break;
            }
            cout<<" Имате грешка! Въведете номер на ново: ";
            cin>>number;
        }
    }
}
}

```

Когато се открие човека със съответния номер се активира под функцията deposits(). В нея се преглежда броя на влоговете, който има дадения човек, изкарва се съответното съобщение в зависимост от броя на влоговете и вида им и се дава възможност, ако има повече от един да си избере от къде да тегли.

Ако има 1 влог:

```

void deposits(person p[],int h) // Подфункция за функция withdrawal_deposit().
{
    int money;
    if(p[h].counts==1) // Ако има 1 влог.
    {
        cout<<" Вие имате 1 влог на стойност "<<p[h].dp[0].amount<<" а "<<p[h].dp[0].currency<<endl;
        if(p[h].dp[0].amount==0) // Ако няма пари в сметката.
            cout<<" Вашия влог е празен! Не можете да теглите пари!"<<endl;
        else // Ако не е.
        {
            cout<<" Въведете желаната сума на теглене: ";
            cin>>money;
            while (money>p[h].dp[0].amount)
            { // Ако въведената сума е по-голяма от текущата.
                cout<<" Нямате толкова пари в сметката:"<<endl;
                string s;
                int k=0;
                do{ // Проверка дали желае да въведе нова сума.
                    cout<<" Желаете ли да въведете нова сума? (да/не) ";
                    cin>>s;
                    if(s=="да")
                    {
                        k=1;
                        cout<<" Въведете нова сума за теглене: ";
                        cin>>money;
                    }
                    else if(s=="не")
                    {
                        money=0;
                        break;
                    }
                    else
                        cout<<" Имате грешка!"<<endl;
                }while(k!=1);
            }
            if(money<=p[h].dp[0].amount)
            {
                p[h].dp[0].amount=p[h].dp[0].amount-money;
            }
        }
    }
}

```

Ако има 2 влога се изкарва питане, от кой влог да се тегли сума.

```
else if(p[h].counts==2) // Ако има 2 влог.
{
    cout<<" Вие имате 2 влога на стойност "<<p[h].dp[0].amount<<" $ "<<p[h].dp[0].currency<<" и "<<p[h].dp[1].amount<<" $ "<<p[h].dp[1].currency<<endl;
    int y;
    do{
        cout<<"\n Изберете влог от който искате да теглите (1/2): ";
        cin>>y;
    }while(y<1 || y>2);
}
```

Ако сме избрали да теглим от влог 1:

```
if (y==1) // Ако теглим от влог 1.
{
    if(p[h].dp[0].amount==0) // Ако няма пари в сметката.
        cout<<" Вашия влог е празен! Не можете да теглите пари!"<<endl;
    else // Ако не е.
    {
        cout<<" Въведете желаната сума на теглене: ";
        cin>>money;
        while (money>p[h].dp[0].amount)
        { // Ако въведената сума е по-голяма от текущата.
            cout<<" Нямаме толкова пари в сметката:"<<endl;
            string s;
            int k=0;
            do{ // Проверка дали желае да въведе нова сума.
                cout<<" Желаете ли да въведете нова сума? (да/не) ";
                cin>>s;
                if (s=="да")
                {
                    k=1;
                    cout<<" Въведете нова сума за теглене: ";
                    cin>>money;
                }
                else if (s=="не")
                {
                    money=0;
                    break;
                }
                else
                    cout<<" Имате грешка!"<<endl;
            }while(k!=1);
        }
        if (money<=p[h].dp[0].amount)
        {
            p[h].dp[0].amount=p[h].dp[0].amount-money;
        }
    }
}
```

Ако сме избрали да теглим от влог 2:

```
else // Ако теглим от влог 2.
{
    if(p[h].dp[1].amount==0) // Ако няма пари в сметката.
        cout<<" Вашия влог е празен! Не можете да теглите пари!"<<endl;
    else // Ако не е.
    {
        cout<<" Въведете желаната сума на теглене: ";
        cin>>money;
        while (money>p[h].dp[1].amount)
        { // Ако въведената сума е по-голяма от текущата.
            cout<<" Нямате толкова пари в сметката:"<<endl;
            string s;
            int k=0;
            do{ // Проверка дали желаете да въведете нова сума.
                cout<<" Желаете ли да въведете нова сума? (да/не) ";
                cin>>s;
                if (s=="да")
                {
                    k=1;
                    cout<<" Въведете нова сума за теглене: ";
                    cin>>money;
                }
                else if (s=="не")
                {
                    money=0;
                    break;
                }
                else
                    cout<<" Имате грешка!"<<endl;
            }while(k!=1);
        }
        if(money<=p[h].dp[1].amount)
        {
            p[h].dp[1].amount=p[h].dp[1].amount-money;
        }
    }
}
```

Ако има 3 влога се изкарва питане, от кой влог да се тегли сума.

Couta е прекалено голям и не може да се побере на една снимка, той е просто изреждане на 3те влога и тяхната стойност.

```
else // Ако има 3 влога.
{
    cout<<" Вие имате 3 влога на стойност "<<p[h].dp[0].amount<<" в "<<p[h].dp[0].currency<<"; "<<endl;
    int y;
    do{
        cout<<"\n Изберете влог от който искате да теглите (1/2/3): ";
        cin>>y;
    }while(y<1 || y>3);
}
```

Ако сме избрали да теглим от влог 1:

```
if (y==1) // Ако теглим от влог 1.
{
    if(p[h].dp[0].amount==0) // Ако няма пари в сметката.
        cout<<" Вашия влог е празен! Не можете да теглите пари!"<<endl;
    else // Ако не е.
    {
        cout<<" Въведете желаната сума на теглене: ";
        cin>>money;
        while (money>p[h].dp[0].amount)
        { // Ако въведената сума е по-голяма от текущата.
            cout<<" Нямате толкова пари в сметката:"<<endl;
            string s;
            int k=0;
            do{ // Проверка дали желае да въведе нова сума.
                cout<<" Желаете ли да въведете нова сума? (да/не) ";
                cin>>s;
                if(s=="да")
                {
                    k=1;
                    cout<<" Въведете нова сума за теглене: ";
                    cin>>money;
                }
                else if(s=="не")
                {
                    money=0;
                    break;
                }
                else
                    cout<<" Имате грешка!"<<endl;
            }while(k!=1);
        }
        if(money<=p[h].dp[0].amount)
        {
            p[h].dp[0].amount=p[h].dp[0].amount-money;
        }
    }
}
```

Ако сме избрали да теглим от влог 2:

```
else if (y==2) // Ако теглим от влог 2.
{
    if(p[h].dp[1].amount==0) // Ако няма пари в сметката.
        cout<<" Вашия влог е празен! Не можете да теглите пари!"<<endl;
    else // Ако не е.
    {
        cout<<" Въведете желаната сума на теглене: ";
        cin>>money;
        while (money>p[h].dp[1].amount)
        { // Ако въведената сума е по-голяма от текущата.
            cout<<" Нямате толкова пари в сметката:"<<endl;
            string s;
            int k=0;
            do{ // Проверка дали желае да въведе нова сума.
                cout<<" Желаете ли да въведете нова сума? (да/не) ";
                cin>>s;
                if(s=="да")
                {
                    k=1;
                    cout<<" Въведете нова сума за теглене: ";
                    cin>>money;
                }
                else if(s=="не")
                {
                    money=0;
                    break;
                }
                else
                    cout<<" Имате грешка!"<<endl;
            }while(k!=1);
        }
        if(money<=p[h].dp[1].amount)
        {
            p[h].dp[1].amount=p[h].dp[1].amount-money;
        }
    }
}
```

Ако сме избрали да теглим от влог 3: и с това приключва под функцията.

```
else // Ако теглим от влог 3.
{
    if(p[h].dp[2].amount==0) // Ако няма пари в сметката.
        cout<<" Вашия влог е празен! Не можете да теглите пари!"<<endl;
    else // Ако не е.
    {
        cout<<" Въведете желаната сума на теглене: ";
        cin>>money;
        while (money>p[h].dp[2].amount)
        { // Ако въведената сума е по-голяма от текущата.
            cout<<" Нямате толкова пари в сметката:"<<endl;
            string s;
            int k=0;
            do{ // Проверка дали желае да въведе нова сума.
                cout<<" Желаете ли да въведете нова сума? (да/не) ";
                cin>>s;
                if(s=="да")
                {
                    k=1;
                    cout<<" Въведете нова сума за теглене: ";
                    cin>>money;
                }
                else if(s=="не")
                {
                    money=0;
                    break;
                }
                else
                    cout<<" Имате грешка!"<<endl;
            }while(k!=1);
        }
        if(money<=p[h].dp[2].amount)
        {
            p[h].dp[2].amount=p[h].dp[2].amount-money;
        }
    }
}
}
```

b. Въвежда се номер на вложител и се извличат всичките му влогове, а профила му се закрива;

Ако от менюто сме избрали 2 се изпълнява `elsa`. Въвежда се номера на карта на вложителя, като ако не е намерен такъв човек в системата се изкарва грешка. Ако намери такъв, изтрива от системата неговите данни. И извличат неговата сума от влоговете му, която се връчва на човека в лева. С това приключва функцията `withdrawal_deposit(p,i)`.

```
else // Ако сме избрали опция 2.
{
    cout<<" Въведете номера на вложителя: ";
    cin>>number;
    cout<<endl;
    while(br==0)
    {
        for (int x=0; x<i; x++)
        { // Цикъл, който върти елементите на масива.
            if(p[x].card_number==number)
            { // Намиране на номера в системата.
                int suma=0;
                for(int l=0;l<p[x].counts;l++)
                { // Извличане на сумата, която притичава в лева.
                    if(p[x].dp[l].currency=="EURO")
                        suma=suma+p[x].dp[l].amount*1.96;
                    else if(p[x].dp[l].currency=="USD")
                        suma=suma+p[x].dp[l].amount*1.61;
                    else
                        suma+=p[x].dp[l].amount;
                }
                i=i-1;
                for(int k=x; k<i; k++)
                {
                    p[k]=p[k+1];
                }
                br++;
                cout<<" Профилът беше закрит успешно!"<<endl;
                cout<<" Сумата, която извлечена е "<<suma<<" лева."<<endl;
                break;
            }
        }
    }
    if (br==0) // Ако не е намерен в системата.
    {
        if(i==0) // Ако базата данни е празна.
        {
            cout<<" Базата данни е празна."<<endl;
            break;
        }
        cout<<" Имате грешка! Въведете номер на ново: ";
        cin>>number;
    }
}
}
```

Упътване за употреба на подточка Е

След като сме селектирали от главното меню опцията за изтегляне на влог, излиза подменю от където може да изберем дали да теглим сума от влог, или да прекратим профил. При избиране на първата опция, ще се изведе съобщение за въвеждане на номера на вложителя. Ако въведем номер, който не съществува се изкарва грешка, а ако няма записи в програмата се изкарва съответното съобщение.

```
МЕНЮ
1. Теглене на сума от влог
2. Прекратяване на профила
Въведете избор [1-2]:1
```

```
Въведете номера на вложителя: 141531
Имате грешка! Въведете номер на ново: 241531
```

```
Въведете номера на вложителя: 41531
Базата данни е празна.
```

Ако се въведе правилен номер се изкарват влоговете на човека и тяхната сума. Дава се възможност да си избере, от къде да тегли пари. Когато избере влог, се въвежда желаната сума. Ако тя е по-голяма от количеството, което има в сметката се извежда съответното съобщение.

```
Въведете номера на вложителя: 526574

Вие имате 3 влога на стойност 2140 в BGN, 550 в EURO и 670 в USD

Изберете влог от който искате да теглите (1/2/3): 4

Изберете влог от който искате да теглите (1/2/3): 1
Въведете желаната сума на теглене: 2500
Нямате толкова пари в сметката:
Желаете ли да въведете нова сума? (да/не) да
Въведете нова сума за теглене: 2140
```

Човек може да изтегли цялата сума от даден влог, но при неговото изпразване, той не се закрива. Ако се опита да тегли от подобен се изкарва съответното съобщение.

```
Въведете номера на вложителя: 526574

Вие имате 3 влога на стойност 0 в BGN, 550 в EURO и 670 в USD

Изберете влог от който искате да теглите (1/2/3): 1
Вашия влог е празен! Не можете да теглите пари!
```

Ако от подменюто изберем опцията за прекратяване на профил, ще ни изведе съобщение да въведем номер. Отново ако въведем грешен номер ще изведе грешка. Когато въведем правилен номер, сметката на човека се закрива, а сумата му бива изтеглена и преобразувана в лева.

```
Въведете номера на вложителя: 226574

Имате грешка! Въведете номер на ново: 526574
Профилът беше закрит успешно!
Сумата, която извлечена е 4296 лева.
```


Ако от главното меню изберем опцията за извеждане на всички вложители ще видим, че човекът с номер 526574 вече не съществува.

```
Всички вложители:
Номер: 241531
Три имена: Иван Сотиров Тодоров
Брой влогове: 1

Влог №1: BGN
Сума: 2415

Номер: 241521
Три имена: Мария Колева Романова
Брой влогове: 2

Влог №1: EURO
Сума: 3520

Влог №2: BGN
Сума: 1000
```

4. Допълнение Трето

Г. Справки за влоговете с под меню

За изпълняването на тази част от задачата създавам функцията `references(p,i)`, която е от тип `void`, тъй като само извършва действия. Функцията използва входни данни за масива `p`, в който е запазена информацията за въведените вложители, а променливата `i` показва броя на записите в масива. Функцията е съставена от първоначално от меню, в което се избира дали да се изведат вложителите в подреден ред, да се изведе вложителя с най-голям влог от въведена валута, или да се изведат вложителите, които имат само един влог. След това се проверява дали базата от данни е празна, ако е се изкарва съответното съобщение, ако не в зависимост от избора се изпълнява `switch`, който се обръща към съответната под функция.

```
void references(person p[],int i)
{
    int n;
    cout<<"\n\t МЕНЮ";
    cout<<"\n 1. Извеждане на вложителите в подреден ред";
    cout<<"\n 2. Извеждане на вложителя с най-голям влог от въведена валута";
    cout<<"\n 3. Извеждане на вложителите, които имат само един влог";
    do{// С помощта на цикъл извеждаме нашия избор, който задължително трябва да е от 1-3
        // ако не е в този диапазон трябва потребителя да въведе нова стойност.
        cout<<"\n Въведете избор [1-3]:";
        cin>>n;
    }while(n<1 || n>3);
    system("cls");
    if (i==0)
    { // Ако в системата няма въведени записи.
        cout<<" Имате грешка, в системата няма въведени вложители."<<endl<<endl;
    }
    else
    {
        switch (n)
        {
            case 1: sort_depositors(p,i); break; // Използване на подфункция.
            case 2: max_deposit(p,i); break; // Използване на подфункция.
            case 3: depositors_one(p,i); // Използване на подфункция.
        }
    }
}
```

а. Извеждане на вложителите в подреден ред по:

Ако изберем от менюто 1, ще се изпълни под функцията `sort_depositors(p, i)`. Функцията се състои от подменю, в което се избира дали вложителите да се подредят по датата на основаване на влоговете, или по общата стойност на влоговете.

```
void sort_depositors(person p[],int i) // Подфункция за функция references().
{
    int br=0;
    int n;
    cout<<"\n Извеждане на вложителите в подреден ред по:";
    cout<<"\n 1. Датата на основаване на влоговете.";
    cout<<"\n 2. Общата стойност на влоговете.";
    do{ // С помощта на цикъл въвеждаме нашия избор, който задължително трябва да е от 1-2
        // ако не е в този диапазон трябва потребителя да въведе нова стойност.
        cout<<"\n Въведете избор [1-2]:";
        cin>>n;
    }while(n<1 || n>2);
    system("cls");
}
```

і. Датата на основаване на влоговете;

При избиране на опция 1 се активира `if`, при който първоначално се създава нова структура, и две нови променливи, които помагат за запазването на информацията.

```
if (n==1) // Ако сме избрали избор 1.
{
    struct depositor
    { // Структура която запазва номера на карта, името, и датата на създаване.
        unsigned int card_number;
        string name;
        string day;
        string month;
        string year;
    }d[50],d2[50]; // Създаване на два масива от по типа на структурата.
}
```

Чрез помощта на `for` информацията от главната структура на програмата се прехвърля в подструктура с цел да не се промени информацията в главната структура.

```
for (int n=0; n<i; n++)
{ // Чрез цикъл се запазва нужната информация от главния цикъл.
    d[n].card_number=p[n].card_number;
    d[n].name=p[n].name;
    d[n].day=p[n].day;
    d[n].month=p[n].month;
    d[n].year=p[n].year;
}
```

След това с помощта на два `for`a извършвам метода на мехурчето, благодарение на който подреждам влоговете по деня им на създаване в низходящ ред. Първия `for` тръгва от края на елементите и се върти докато не стане 1, а втория от началото им и се върти докато не стигне елемента въведен в първия `for`.

С помощта на функция от библиотеката sstream стринга се превръща в число, тъй като въведената дата е от тип стринг. После първоначално сравнявам годините на елемента с този след него. Ако е по-малък го сменя с този след него. По този начин информацията се подрежда по-низходящ ред.

```
for (int n=i; n>1; n--)
{ // Цикъл който се върти от края на елементите докато стане 1.
  for(int j=0; j<n-1; j++)
  { // Цикъл който се върти от началото на елементите до елемента от горния цикъл.
    // Превръщане на стринга в число.
    stringstream a(d[j].year);
    int aa;
    a>>aa;
    stringstream b(d[j+1].year);
    int bb;
    b>>bb;
    if (aa<bb) // Сравняване на годините.
    {
      d2[j]=d[j];
      d[j]=d[j+1];
      d[j+1]=d2[j];
    }
    else if(aa==bb) // Сравняване на годините.
```

Ако годините са равни, тогава се проверяват месеците, като отново с помощта на функция от библиотеката sstream стринга се превръща в число, тъй като въведената дата е от тип стринг. Ако предходния елемент е по-малък го сменя с този след него. По този начин информацията се подрежда по-низходящ ред.

```
else if(aa==bb) // Сравняване на годините.
{ // Превръщане на стринга в число.
  stringstream c(d[j].month);
  int cc;
  c>>cc;
  stringstream l(d[j+1].month);
  int ll;
  l>>ll;
  if(cc<ll) // Сравняване на месеците.
  {
    d2[j]=d[j];
    d[j]=d[j+1];
    d[j+1]=d2[j];
  }
  else if(cc==ll) // Сравняване на месеците.
```

1. $\frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$ (Probability of getting two heads)

1. *Journal of Management Studies*, 1997, 34, 1, 1-14.

[illegible]

10

Чрез помощта на for информацията от главната структура на програмата се прехвърля в подструктура с цел да не се промени информацията в главната структура.

```
for (int n=0; n<i; n++)
{ // Чрез цикъл се запазва нужната информация от главния цикъл.
    int sum=0;
    int x=p[n].counts;
    for(int y=0; y<x; y++)
        sum=sum+p[n].dp[y].amount;
    d[n].card_number=p[n].card_number;
    d[n].name=p[n].name;
    d[n].amount=sum;
}
```

След това с помощта на два for-a извършвам метода на мехурчето, благодарение на който подреждам влоговете по общата сума в низходящ ред. Първия for тръгва от края на елементите и се върти докато не стане 1, а втория от началото им и се върти докато не стигне елемента въведен в първия for.

Сравнявам общата сума на вложителя с този след него в системата. Ако е по-малък си сменят местата. По този начин информацията се подрежда по-низходящ ред.

```
for (int n=i; n>1; n--)
{ // Цикъл който се върти от края на елементите докато стане 1.
    for(int j=0; j<n-1; j++)
    { // Цикъл който се върти от началото на елементите до елемента от горния цикъл.
        if(d[j].amount<d[j+1].amount) // Сравняване на сумите.
        {
            d2[j]=d[j];
            d[j]=d[j+1];
            d[j+1]=d2[j];
        }
    }
}
```

Накрая се извежда запазената подредена информация в низходящ ред спрямо общата стойност на влоговете. Като информацията, която се отпечатва е номера на картата, името на вложителя и общата сума на влоговете му и с това свършва функцията.

```
cout<<" Извеждане на вложителите подред по общата стойност на влоговете."<<endl;
for (int n=0; n<i; n++)
{ // Извеждане на подредения масив.
    cout<<"\t №"<<n+1<<endl;
    cout<<" Номер: "<<d[n].card_number<<endl;
    cout<<" Име: "<<d[n].name<<endl;
    cout<<" Сума: "<<d[n].amount<<endl<<endl;
}
}
```

б. Извеждане на вложителя с най-голям влог от въведена валута;

Ако изберем от менюто за справки опцията за извеждане на вложителя с най-голям влог от въведена валута, ще се изпълни под функцията `max_deposit(p,i)`. В началото се изкарва съобщение, което пита за избора на валута, която задължително трябва да е BGN/USD/EURO. След това с помощта на `for` се преглежда базата от данни. Като с помощта на `if` се разглеждат само тези, от посочената валута, като се запазва този с най-голямата сума. Ако базата данни е празна се изкарва съответното съобщение.

```
void max_deposit(person p[],int i) // Подфункция за функция references().
{
    string s,name;
    int k, maxa=0,number,br=0;
    do{ // Попроса за правилно въведена валута.
        cout<<" Изберете валута(BGN (лева), USD (долара),EURO (евро)): ";
        cin>>s;
        if(s=="BGN" || s=="USD" || s=="EURO")
            k=1;
        else
            cout<<" Имате грешка!"<<endl;
    }while(k!=1);
    system("cls");
    for (int n=0; n<i; n++)
    { // Въртене на масива.
        for (int x=0; x<3; x++)
        { // Въртене на подмасива.
            if(p[n].dp[x].currency==s)
            {
                if (p[n].dp[x].amount>maxa) // Сравняване на сумата.
                {
                    maxa=p[n].dp[x].amount;
                    number=p[n].card_number;
                    name=p[n].name;
                }
            }
            br++;
        }
    }
    if (br==0) // Ако няма хора с подобни влогове.
        cout<<" Няма вложители с влог "<<s<<"."<<endl<<endl;
    else // Ако има изкарвава човека с най-голям.
        cout<<"\n Вложителя, който има най-голяма сума на стойност "<<maxa<<" в "<<s<<" е "<<name<<" с номер на картата "<<number<<"."<<endl;
}
```

с. Извеждане на вложителите, които имат само един влог.

Ако изберем от менюто за справки опцията за извеждане на вложителите, които имат само един влог, ще се изпълни под функцията `depositors_one(p,i)`. Функцията е кратка, като тя се състои от един `for`, чрез който се преглежда базата от данни. След това с помощта на `if` се изкарват само хората който имат един влог. Ако базата данни е празна се изкарва съответното съобщение.

```
void depositors_one(person p[],int i) // Подфункция за функция references().
{
    int br=0;
    cout<<"\t Вложителите, които имат само един влог:"<<endl;
    for (int n=0; n<i; n++)
    { // Въртене на масива.
        if(p[n].counts==1)
        { // Изкарване на хората с един влог.
            cout<<"\n Номер: "<<p[n].card_number<<endl;
            cout<<" Три имена: "<<p[n].name<<endl;
            cout<<" Влог: "<<p[n].dp[0].currency<<endl;
            cout<<" Сума: "<<p[n].dp[0].amount<<endl;
            br++;
        }
    }
    if (br==0) // Ако няма такива хора.
        cout<<" Няма такива вложители."<<endl<<endl;
}
```

Упътване за употреба на подточка F

След като сме селектирали от главното меню опцията за справки, излиза подменю от където може да изберем дали да изведем вложителите в подреден ред, да изведем вложителя с най-голям влог от въведена валута, или да изведем вложителите, които имат само един влог.

```
МЕНЮ
1. Извеждане на вложителите в подреден ред
2. Извеждане на вложителя с най-голям влог от въведена валута
3. Извеждане на вложителите, които имат само един влог
Въведете избор [1-3]:1
```

Ако изберем опция, а базата от данни е празна, ще се изведе съответното съобщение и ще се върнем обратно на главното меню.

```
Имате грешка, в системата няма въведени вложители.

МЕНЮ
1. Добавяне на вложител
2. Извеждане на всички вложители
3. Добавяне на нов влог
4. Изтегляне на влог
5. Справки
6. Запазване на въведените данни във файл
7. Прехвърляне на данните от файла в програмата
8. Извеждане на данните от файла
9. Изчистване на запазените данни във файла
10. Край
Въведете избор [1-10]:
```

При избиране на първата опция от подменюто, ще се изведе второ подменю, от където можем да изберем дали да подредим вложителите по дата на основаване на влоговете, или по общата им стойност.

```
Извеждане на вложителите в подреден ред по:
1. Датата на основаване на влоговете;
2. Общата стойност на влоговете.
Въведете избор [1-2]:1
```

При избиране на първата опция, ще се изведат вложителите в подреден ред по дата. Това трябва да се изведе ако използваме въведената информация от В подточка.

```
Извеждане на вложителите подред по датата на основаване на влоговете.
*****
Номер: 241531
Име: Иван Сотиров Тодоров
Дата на основаване: 15/11/2020
*****
Номер: 241521
Име: Мария Колева Романова
Дата на основаване: 25/02/2019
*****
Номер: 526574
Име: Стоян Петров Генчев
Дата на основаване: 21/05/2018
*****
```


При избиране на втората опция, ще се изведат вложителите в подреден ред по общата им сума. Това трябва да се изведе ако използваме въведената информация от В подточка и от D подточка.

Извеждане на вложителите подред по общата стойност на влоговете.

№1

Номер: 241521

Име: Мария Колева Романова

Сума: 3520

№2

Номер: 526574

Име: Стоян Петров Генчев

Сума: 3360

№3

Номер: 241531

Име: Иван Сотиров Тодоров

Сума: 2415

Ако от първото подменю изберем опция 2, за извеждане на най-големия влог от въведена валута, първо ще ни излезе запитване за желаната валута.

Изберете валута(BGN (лева), USD (долара),EURO (евро)): BGN_

Ако няма въведени данни в системата, ще се изведе съответното съобщение.

Няма вложители с влог BGN.

Ако съществуват хора с влог в съответната валута, ще се изкара този с най-големия влог.

Вложителя, който има най-голяма сума на стойност 2415 в BGN е Иван Сотиров Тодоров с номер на картата 241531.

Ако от подменюто изберем опция 3, за извеждане на вложителите с един влог и няма такива ще се изведе съответното съобщение.

Вложителите, които имат само един влог:
Няма такива вложители.

Ако използваме въведената информация от В подточка и от D подточка ще се изведе следното:

Вложителите, които имат само един влог:

Номер: 241531

Три имена: Иван Сотиров Тодоров

Влог: BGN

Сума: 2415

Номер: 241521

Три имена: Мария Колева Романова

Влог: EURO

Сума: 3520

G. Данните в програмата да могат да се запазват във файл между две стартирания на програмата.

За изпълняването на тази част от задачата създавам няколко функции, които извършват различни задачи: запазване на въведените данни във файл – `save(p,i)`; прехвърляне на данните от файла в програмата – `fileToProgram(p,i)`; извеждане на данните от файла – `file()`; изчистване на запазените данни във файла `clear_file()`. Всичките функции са от тип `void`, тъй като само извършва действия. Някои от тях използват входни данни за масива `p`, в който е запазена информацията за въведените вложители, а променливата `i` показва броя на записите в масива. Четирите функции служат за по-лесна работа с файла.

Първата функция, за запазване на въведените данни във файл, е съставена от първоначално от отваряне на файла с цел да се изтрие негова информация, ако има такава, за да не стане натрупване на информация. След това файла се отваря на ново и един след друг се запазват елементите от главната структура с помощта на `for`.

```
void save(person p[],int i)
{
    cout<<"\tфайлът беше записан успешно!"<<endl;
    fstream f;
    // Предварително отваряне на файла, с цел да се изтрият предишните данни.
    f.open("Depositors.txt", ios::out | ios::trunc);
    f.close();
    // Отваряне на файла.
    f.open("Depositors.txt",ios::out | ios::in | ios::app);
    f<<i<<endl;
    for (int n=0; n<i; n++)
    {
        int x=0;
        f<<p[n].card_number<<endl;
        f<<p[n].name<<endl;
        f<<p[n].day<<endl;
        f<<p[n].month<<endl;
        f<<p[n].year<<endl;
        f<<p[n].counts<<endl;
        while(x<p[n].counts)
        {
            f<<p[n].dp[x].currency<<endl;
            f<<p[n].dp[x].amount<<endl;
            x++;
        }
    }
    // Затваряне на файла.
    f.close();
}
```

Втората функция, за прехвърляне на данните от файла в програмата, е съставена първоначално от if, който проверява дали функцията е била използвана един път, с цел да не стане грешка и да се натрупа информация. Ако до момента функцията не е била ползвана, се изпълнява else if, който проверява дали файла е празен, ако не е се прехвърлят данните от файла, като се вземат ред по ред.

```
void fileToProgram(person p[],int& i,int& FP)
{
    fstream f;
    f.open("Depositors.txt", ios::out | ios::in);
    // Проверка дали файлът вече е бил прехвърлен.
    if(FP==1)
    {
        cout<<"\tфайлът вече беше прехвърлен."<<endl;
    }
    // Проверка дали файлът е празен.
    else if ( f.peek() == ifstream::traits_type::eof() )
    {
        cout<<" файлът е празен!"<<endl;
    }
    else
    {
        cout<<"\tДанните от файла бяха прехвърлени успешно!"<<endl;
        int j;
        f>>j;
        int n=i+j;
        for(i; i<n; i++)
        {
            f>>p[i].card_number;
            f.ignore();
            getline(f, p[i].name);
            f>>p[i].day;
            f>>p[i].month;
            f>>p[i].year;
            f>>p[i].counts;
            int x=0;
            while (x<p[i].counts)
            {
                f>>p[i].dp[x].currency;
                f>>p[i].dp[x].amount;
                x++;
            }
            FP++;
        }
        f.close();
    }
}
```

Третата функция, за извеждане на данните от файла, е съставена първоначално от if, който проверява дали файла е празен, ако не е информацията от файла се извежда на екрана по определен начин.

```
void file()
{
    fstream f;
    cout<<"\tДанните от файла"<<endl;
    f.open("Depositors.txt", ios::out | ios::in);
    string a,b,c;
    int j;
    j=0;
    // Проверка дали файлът е празен.
    if ( f.peek() == ifstream::traits_type::eof() )
    {
        cout<<" файлът е празен!"<<endl;
    }
    else
    {
        f>>j;
        for(int n=0; n<j; n++)
        {
            f>>a;
            cout<<" Номер: "<<a<<endl;
            f.ignore();
            getline(f, a);
            cout<<" Три имена: "<<a<<endl;
            f>>a;
            f>>b;
            f>>c;
            cout<<" Дата на откриване: "<<a<<"/"<<b<<"/"<<c<<endl;
            int y;
            f>>y;
            cout<<" Брой влогове: "<<y<<endl<<endl;
            int x=0;
            while (x<y)
            {
                f>>a;
                cout<<" Влог №"<<x+1<<": "<<a<<endl;
                f>>a;
                cout<<" Сума: "<<a<<endl<<endl;
                x++;
            }
            cout<<endl;
        }
    }
    f.close();
}
```

Четвъртата функция, за изчистване на запазените данни във файла, е съставена от отваряне на файла чрез режима за изтриване на съдържанието при отваряне, след което файла просто се затваря.

```
void clear_file()
{
    cout<<"\tфайлът беше изчистен успешно!"<<endl;
    fstream f;
    f.open("Depositors.txt", ios::out | ios::trunc);
    f.close();
}
```

Упътване за употреба на подточка G

След като сме селектирали от главното меню опцията за запазване на въведените данни във файл, излиза съобщение, което ни оповестява.

Файлът беше записан успешно!

После като изберем от главното меню опцията за извеждане на данните от файла, ще видим, че данните, които сме въвели от подточка B и подточка D са се записали успешно.

```
Данните от Файла
*****
Номер: 526574
Три имена: Стоян Петров Генчев
Дата на откриване: 21/05/2018
Брой влогове: 3

Влог №1: BGN
Сума: 2140
Влог №2: EURO
Сума: 550
Влог №3: USD
Сума: 670
*****
Номер: 241531
Три имена: Иван Сотиров Тодоров
Дата на откриване: 15/11/2020
Брой влогове: 1

Влог №1: BGN
Сума: 2415
*****
Номер: 241521
Три имена: Мария Колева Романова
Дата на откриване: 25/02/2019
Брой влогове: 1

Влог №1: EURO
Сума: 3520
*****
```

След като run-нем програмата на ново и изберем от менюто прехвърляне на данните от файла в програмата, ще ни излезе съобщение, че данните са били записани успешно.

Данните от файла бяха прехвърлени успешно!

Ако се опитаме повторно да изпълним действието, ще ни изведе съобщение, че вече един път сме прехвърлили данните.

Файлът вече беше прехвърлен.

При избиране на опцията за показване на всички вложители от главното меню, ще видим че данните са прехвърлени успешно.

```
Всички вложители:
*****
Номер: 526574
Три имена: Стоян Петров Генчев
Брой влогове: 3

Влог №1: BGN
Сума: 2140
Влог №2: EURO
Сума: 550
Влог №3: USD
Сума: 670
*****

Номер: 241531
Три имена: Иван Сотиров Тодоров
Брой влогове: 1

Влог №1: BGN
Сума: 2415
*****

Номер: 241521
Три имена: Мария Колева Романова
Брой влогове: 1

Влог №1: EURO
Сума: 3520
*****
```

При избиране на опцията за изчистване на запазените данни във файла от главното меню, ще ни излезе съобщение.

```
Файлът беше изчистен успешно!
```

След това ако пробваме да изведем данните от файла, ще ни излезе следното съобщение.

```
Данните от Файла
Файлът е празен!
```

Ако пък се пробваме да прехвърлим данните от файла в програмата, ще излезе следното съобщение.

```
Файлът е празен!
```

Програмата приключва с функцията `exit()`, която се активира при избиране на опцията край от главното меню.